

**IMPLEMENTASI KONTROL LAMPU DAN KIPAS SECARA  
IMPLISIT MENGGUNAKAN SUARA DENGAN METODE *TEXT*  
*PROCESSING* BERBASIS *EMBEDDED SYSTEM***

**SKRIPSI**

**KEMINATAN TEKNIK KOMPUTER**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Wildo Satrio  
NIM: 135150300111067



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

IMPLEMENTASI KONTROL LAMPU DAN KIPAS SECARA IMPLISIT MENGGUNAKAN  
SUARA DENGAN METODE *TEXT PROCESSING* BERBASIS *EMBEDDED SYSTEM*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Wildo Satrio

NIM: 135150300111067

Skripsi ini telah diuji dan dinyatakan lulus pada  
2 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dahnial Syauqy, S.T., M.T., M.Sc.

NIK: 2016078704231002

Hurriyatul Fitriyah, S.T., M.Sc

NIP: 19851001 201504 2 003

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D

NIP. 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2018



Wildo Satrio

NIM: 135150300111067

## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga peneliti dapat menyelesaikan laporan skripsi dengan judul “Implementasi kontrol lampu dan kipas secara implisit menggunakan suara dengan metode *Text Processing* Berbasis *Embedded System*” dapat diselesaikan dengan baik.

Dalam penyusunan dan penelitian skripsi ini tidak lepas dari dukungan serta bantuan moral dan materiil yang telah diberikan dari berbagai pihak, maka peneliti mengucapkan banyak terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Heru Nurwarsito, Ir., M.Kom. selaku Wakil Ketua I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
5. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing satu yang telah memberikan ilmu, saran, penjelasan dan motivasi serta membantu dalam penyusunan laporan penulis.
6. Ibu Hurriyatul Fitriyah, S.T, M.Sc. selaku dosen pembimbing dua yang telah memberikan ilmu, saran, penjelasan dan motivasi kepada penulis.
7. Papa Usharinal dan Mama Hazrilina selaku orang tua yang penulis cintai yang selalu memberi dukungan, dorongan dan do’a agar penulis dapat menyelesaikan skripsi ini dengan lancar.
8. M. Anggun Novembra, M. Rendra Sadinov, Hifzil Waldy Lahuda dan M. Farid Alman Faluthi selaku kakak – kakak dan adik yang selalu memberi dukungan, dorongan serta do’a agar penulis dapat menyelesaikan studinya.
9. Seluruh civitas akademika Informatika Universitas Brawijaya dan terkhusus untuk teman-teman Teknik Komputer Angkatan 2013 yang telah banyak memberi bantuan dan dukungan selama peneliti menempuh studi di Teknik Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
10. Grup kedai 999, grup kost29, grup BellatoUnion, grup kalemp dan seluruh teman-teman dari FILKOM yang tidak dapat disebutkan namanya satu persatu yang telah memberi dukungan dan do’a



11. Seluruh pihak yang tidak dapat diucapkan satu persatu, peneliti mengucapkan banyak terima kasih atas segala bentuk dukungan dan doa sehingga laporan skripsi ini dapat terselesaikan.

Peneliti menyadari bahwa tulisan ini masih jauh dari kata sempurna dan masih memiliki berbagai macam kekurangan. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun, agar ke depannya penulis dapat menjadi lebih baik lagi. Semoga isi Laporan Skripsi ini dapat memberi manfaat bagi perkembangan ilmu pengetahuan di kemudian hari.

Malang, 2 Agustus 2018

Wildo Satrio

wildo.satrio@gmail.com



## ABSTRAK

Teknologi pengenalan suara dapat digunakan sebagai Kontrol berbagai macam peralatan elektronik, diantaranya lampu dan kipas. Bagaimana perangkat – perangkat elektronik dapat mengerti bahasa manusia yang kompleks, Hal ini merupakan suatu tantangan tertentu bagi teknologi saat ini. Berdasarkan masalah tersebut maka dibuat kontrol kipas dan lampu secara implisit menggunakan suara. Arduino sebagai sistem kontrol terhadap lampu dan kipas dengan memanfaatkan Android untuk pengolahan suara dengan metode *text processing*. Data suara yang masuk lewat Android diubah menjadi teks. Setiap teks yang berbentuk kalimat dipisah menjadi satuan kata. Kemudian tiap kata dicocokkan dengan *database* yang telah dibuat sebelumnya menggunakan algoritme *stopword removal wordlist*. Hasil pencocokan tersebut dikirim ke Arduino dan dilakukan pengolahan data untuk menghasilkan *output* berupa kontrol lampu dan kipas. Dari hasil pengujian, sistem ini memiliki nilai keberhasilan 100% pada pengujian fungsional dan pengujian pengolahan *input* suara *User*. Pada pengujian keberhasilan sistem untuk mengolah kalimat implisit *user* menjadi *output* yang diinginkan memiliki akurasi sebesar 96,67%. Selanjutnya untuk pengujian performa aplikasi untuk mengolah kalimat, sistem memiliki rata-rata waktu bertambah 1 detik sesuai banyak kata yang dimasukkan, sedangkan untuk proses pengolahan kalimat bertambah sebesar 200 *nano* detik seiring dengan banyaknya masukan kata.

Kata kunci: Android Device, Arduino, Implisit, Teknologi Suara, Text Processing

## ABSTRACT

Speech recognition technology can be used as a Control of various kinds of electronic equipment, including lights and fans. How electronic devices can understand complex human language, This is a particular challenge for today's technology. Based on the problem then made the control of the fan and the light implicitly using the sound. Arduino as a control system to light and fan by utilizing Android for voice processing with text processing method. Voice data entered via Android is converted to text. Each sentence-shaped text is separated into unit words. Then each word is matched with a database that has been created before using wordword stopword removal algorithm. The matching results are sent to Arduino and the data is processed to produce *output* in the form of light and fan control. From the test results, this system has a 100% success rate on functional testing and User voice *input* processing testing. In testing the success of the system to process the user implicit sentence into the desired *output* has an accuracy of 96.67%. Furthermore, to test the performance of the application to process sentences, the system has an average time of 1 second increase according to the many words entered, while the sentence processing process increases by 200 nano seconds along with the number of *input* words.

**Keywords :** *Android Device, Arduino, Implicit, Text Processing, Speech Recognition*

## DAFTAR ISI

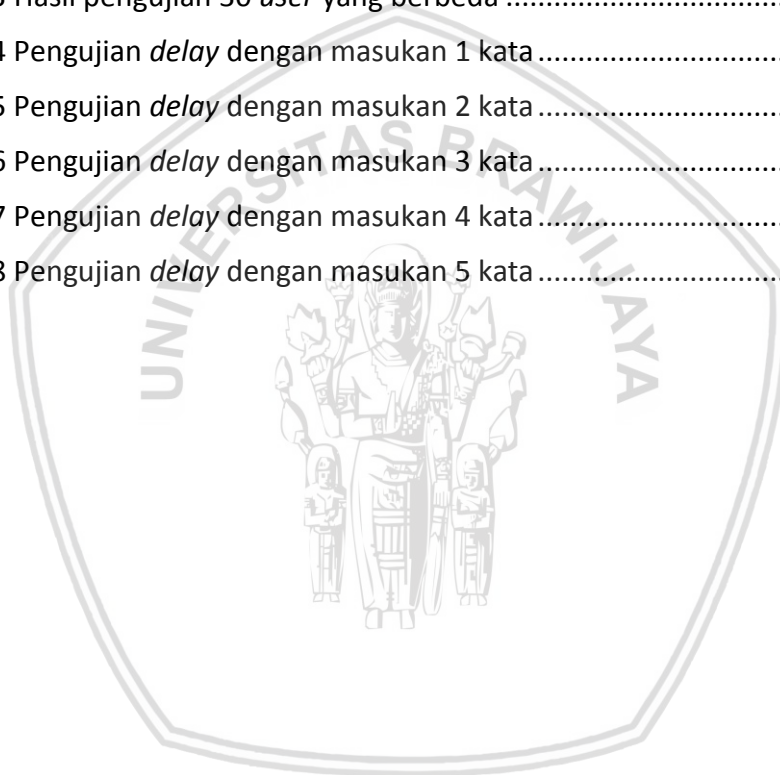
PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	4
2.1 Kajian Pustaka .....	4
2.2 Dasar Teori .....	5
2.2.1 <i>Smart Home</i> .....	5
2.2.2 <i>Text Processing</i> .....	6
2.2.3 Arduino.....	7
2.2.4 HC-05.....	9
2.2.5 <i>Relay 2 Channel Output</i> .....	10
2.2.6 <i>Speech recognition Android Device</i> .....	11
2.2.7 Penghitungan Persentase Akurasi .....	12
BAB 3 METODOLOGI .....	13
3.1 Metodologi Penelitian .....	13
3.2 Analisis kebutuhan.....	13
3.3 Perancangan Sistem.....	14

3.4 Implementasi Sistem .....	14
3.5 Pengujian dan Analisis Sistem .....	15
3.6 Penarikan kesimpulan dan saran .....	15
<b>BAB 4 ANALISIS KEBUTUHAN .....</b>	<b>16</b>
4.1 Gambaran umum sistem .....	16
4.2 Analisis kebutuhan sistem .....	16
4.2.1 Kebutuhan pengguna .....	17
4.2.2 Kebutuhan sistem .....	17
4.3 Kebutuhan fungsional .....	21
4.4 Kebutuhan non fungsional .....	22
4.4.1 Karakteristik yang berbeda tiap individu .....	22
4.4.2 Lingkungan pengoperasian sistem .....	22
4.4.3 Asumsi ketergantungan tiap komponen .....	22
4.4.4 Pengambilan data .....	23
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM .....</b>	<b>24</b>
5.1 Perancangan sistem .....	24
5.1.1 Perancangan Komunikasi antar Sistem .....	25
5.1.2 Perancangan rangkaian pada Arduino .....	26
5.1.3 Perancangan program pada Android .....	29
5.1.4 Perancangan program pada Arduino .....	33
5.2 Implementasi Sistem .....	35
5.2.1 Implementasi komunikasi sistem .....	35
5.2.2 Implementasi sistem pada Android .....	35
5.2.3 Implementasi sistem pada Arduino .....	42
<b>BAB 6 PENGUJIAN DAN ANALISIS .....</b>	<b>45</b>
6.1 Pengujian Fungsional Sistem .....	45
6.1.1 Tujuan pengujian .....	45
6.1.2 Prosedur pengujian .....	45
6.1.3 Hasil Pengujian .....	45
6.1.4 Analisis pengujian .....	49
6.2 Pengujian Proses Pengolahan Teks .....	49
6.2.1 Tujuan Pengujian .....	49

6.2.2 Prosedur Pengujian .....	49
6.2.3 Hasil Pengujian .....	49
6.2.4 Analisis Pengujian.....	54
6.3 Pengujian Sistem Dengan Individu Yang Berbeda .....	56
6.3.1 Tujuan pengujian.....	56
6.3.2 Prosedur pengujian .....	56
6.3.3 Hasil pengujian .....	56
6.4 Pengujian Performa Aplikasi .....	57
6.4.1 Tujuan pengujian.....	57
6.4.2 Prosedur pengujian .....	58
6.4.3 Hasil Pengujian .....	58
6.4.4 Analisis Pengujian.....	60
BAB 7 PENUTUP .....	62
7.1 Kesimpulan.....	62
7.2 Saran .....	62
DAFTAR PUSTAKA.....	63
LAMPIRAN A GAMBAR <i>USER</i> SAAT PENGUJIAN.....	65
LAMPIRAN B KUESIONER <i>DATABASE</i> .....	66

## DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Mega 2560 .....	9
Tabel 4.1 Spesifikasi Android <i>Device</i> yang Digunakan .....	17
Tabel 4.2 Spesifikasi Arduino .....	18
Tabel 4.3 Spesifikasi Komputer yang Digunakan .....	19
Tabel 6.1 Hasil Pengujian Fungsional Sistem .....	47
Tabel 6.2 Hasil Pengujian Pengolahan Teks dan Suara .....	54
Tabel 6.3 Hasil pengujian 30 <i>user</i> yang berbeda .....	56
Tabel 6.4 Pengujian <i>delay</i> dengan masukan 1 kata .....	58
Tabel 6.5 Pengujian <i>delay</i> dengan masukan 2 kata .....	58
Tabel 6.6 Pengujian <i>delay</i> dengan masukan 3 kata .....	59
Tabel 6.7 Pengujian <i>delay</i> dengan masukan 4 kata .....	59
Tabel 6.8 Pengujian <i>delay</i> dengan masukan 5 kata .....	60

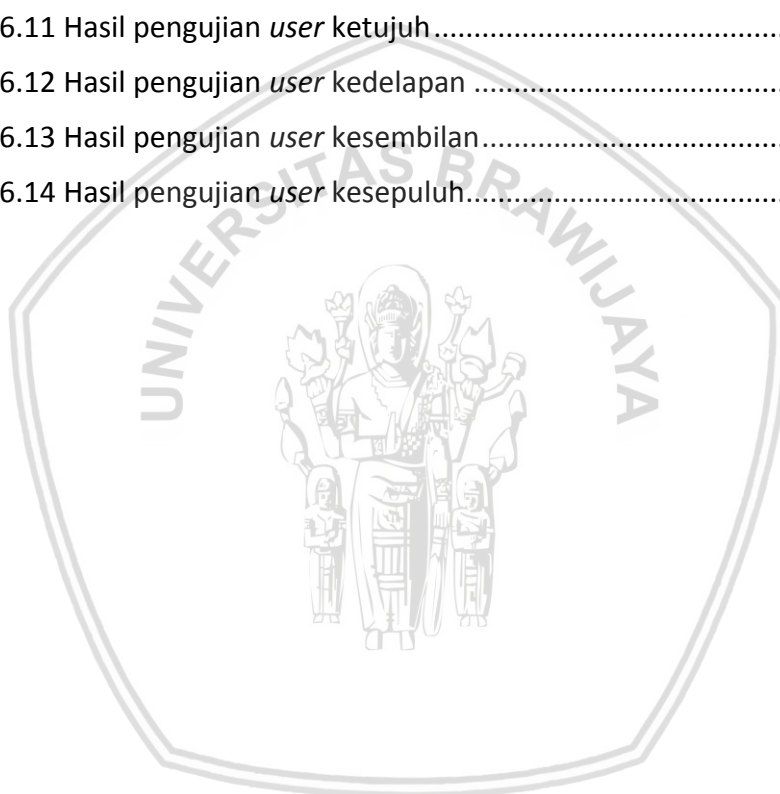




## DAFTAR GAMBAR

Gambar 2.1 Rancangan sistem.....	4
Gambar 2.2 Proses kendali quadcopter menggunakan suara .....	5
Gambar 2.3 <i>Smart Home</i> .....	6
Gambar 2.4 Alur Metode <i>Text Processing</i> .....	7
Gambar 2.5 Arduino Mega 2560.....	8
Gambar 2.6 HC-05 .....	9
Gambar 2.7 <i>Relay 5V DC 2 Channel Output</i> .....	10
Gambar 2.8 Bagian – bagian dari <i>relay</i> .....	11
Gambar 4.1 Analisis kebutuhan sistem.....	16
Gambar 4.2 Kebutuhan <i>Software</i> .....	19
Gambar 4.3 Kebutuhan Komunikasi .....	21
Gambar 5.1 Alur Perancangan Sistem .....	24
Gambar 5.2 Perancangan Komunikasi Program .....	25
Gambar 5.3 Rangkaian HC-05 ke Arduino Mega 2650 .....	26
Gambar 5.4 Rangkaian <i>Relay 2 Channel</i> ke Arduino Mega 2650.....	27
Gambar 5.5 Skema Rangkaian <i>Hardware</i> .....	28
Gambar 5.6 Implementasi <i>hardware</i> .....	43
Gambar 5.7 Sistem tampak luar.....	44
Gambar 5.8 Perancangan database .....	30
Gambar 5.9 <i>Flowchart</i> alur program Android.....	32
Gambar 5.10 Alur pengolahan Teks.....	33
Gambar 5.11 <i>Flowchart</i> alur program pada Arduino.....	34
Gambar 5.12 Implementasi Komunikasi Sistem .....	35
Gambar 5.13 Pembuatan <i>database</i> database.db .....	36
Gambar 5.14 Isi dari tabel <i>Kalimat_inti</i> kolom NAMA dan PERINTAH .....	37
Gambar 5.15 Halaman Awal Program.....	38
Gambar 5.16 Halaman <i>Home</i> .....	40
Gambar 5.17 Implementasi dari pengolahan suara .....	41
Gambar 6.1 Hasil pengujian masukan suara .....	46
Gambar 6.2 Hasil pengujian pemotongan kalimat menjadi kata .....	46

Gambar 6.3 Hasil pencocokan tiap kata pada <i>database</i> di Android.....	46
Gambar 6.4 Pengujian fungsional pada Arduino .....	47
Gambar 6.5 Hasil pengujian <i>user</i> pertama .....	50
Gambar 6.6 Hasil pengujian <i>user</i> kedua .....	50
Gambar 6.7 Hasil pengujian <i>user</i> ketiga .....	51
Gambar 6.8 Hasil pengujian <i>user</i> keempat .....	51
Gambar 6.9 Hasil pengujian <i>user</i> kelima.....	52
Gambar 6.10 Hasil pengujian <i>user</i> keenam .....	52
Gambar 6.11 Hasil pengujian <i>user</i> ketujuh.....	53
Gambar 6.12 Hasil pengujian <i>user</i> kedelapan .....	53
Gambar 6.13 Hasil pengujian <i>user</i> kesembilan.....	54
Gambar 6.14 Hasil pengujian <i>user</i> kesepuluh.....	54



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Berbicara adalah suatu bentuk komunikasi yang paling natural yang dilakukan oleh manusia (Sharmal,2011). Sebagai media komunikasi utama, teknologi pengenalan suara digunakan untuk mengartikan suara yang dikeluarkan manusia menjadi sebuah perintah yang dapat dimengerti oleh mesin. Saat ini banyak teknologi pengenalan suara yang digunakan pada berbagai macam peralatan elektronik. Sehingga kontrol terhadap peralatan elektronik ini lebih efektif daripada menggunakan sistem saklar, Karena sistem saklar ini kurang efektif bagi beberapa kalangan yang mempunyai keterbatasan fisik dan kesibukan yang tinggi, sehingga kontrol melalui suara merupakan suatu solusi untuk melakukan kontrol terhadap perangkat-perangkat elektronik. Bagaimana perangkat – perangkat elektronik dapat mengerti bahasa manusia yang kompleks. Hal ini merupakan suatu tantangan tertentu bagi teknologi saat ini (Vibeke A,2017). Salah satu bahasa manusia yang kompleks ini adalah kalimat implisit. Kalimat implisit menurut KBBI adalah kalimat yang termasuk (terkandung) di dalamnya (meskipun tidak dinyatakan secara jelas atau terang-terangan).

*Internet of Things (IoT)* secara umum, menggambarkan sebuah jaringan koneksi dari perangkat – perangkat elektronik yang biasanya diberi kecerdasan sendiri. *IoT* sendiri menggunakan internet yang menghubungkan satu perangkat dengan perangkat lainnya melalui *embedded systems*, yang akhirnya menghubungkan kesatuan perangkat – perangkat tersebut dengan user. Salah satu penerapan *Internet of Things* adalah sistem monitoring atau kontrol yang menggunakan sensor dan aktuator pada sebuah lingkungan rumah atau yang disebut sebagai *Smart Home*. Lingkungan rumah yang cerdas adalah Sebuah jaringan lokal yang terdiri dari sensor, aktuator, *display*, elemen komputasi yang terhubung dalam sebuah sistem *embedded* ke dalam peralatan rumah tangga (Björkskog,2010).

*Speech recognition* merupakan teknologi komputer masa kini yang digunakan untuk mengidentifikasi suara yang diucapkan oleh seseorang. Implementasi *speech recognition* dapat digunakan untuk berbagai macam keperluan, misalnya perintah suara untuk menjalankan aplikasi komputer, perintah suara untuk mengendalikan perangkat elektronik, dan lain-lain. Cara kerja yang dilakukan oleh sistem dengan mendeteksi tingkat penekanan suara, lalu dicocokkan dengan *database* yang tersedia (Kisumal, 2010). Penggunaan *speech recognition* dapat diterapkan dengan berbagai macam mikrokontroller, seperti Arduino, Raspberry Pi, dan AVR. Pada penelitian ini akan menggunakan mikrokontroller Arduino, Karena perangkat lunaknya mudah dikembangkan dan mampu mencakup semua kebutuhan dalam penelitian. *Smartphone* atau Android *device* digunakan untuk masukan suara dari *user* dan sebagai tempat pengolahan kalimat, sedangkan Arduino sebagai penerima dan mengolah data

dari Android *device*. Hasil proses pada Arduino akan dijadikan perintah untuk menyalakan lampu dan kipas.

Parameter yang akan diuji berupa uji masukan perintah suara secara implisit dengan *user* yang berbeda, dan yang terakhir uji keberhasilan sistem dalam memahami *input* user. Dengan adanya skripsi ini, penulis berusaha mengembangkan teknologi sistem kontrol kipas dan lampu supaya lebih mudah pengaturannya dalam berbagai bidang khususnya melalui suara.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat dirumuskan menjadi rumusan masalah sebagai berikut:

1. Bagaimana menggunakan *speech recognition* Android sebagai kontrol terhadap kipas dan lampu secara implisit menggunakan suara?
2. Bagaimana mengimplementasikan sistem kontrol kipas dan lampu secara implisit menggunakan suara dengan metode *Text Processing*?
3. Bagaimana analisa sistem kontrol kipas dan lampu secara implisit menggunakan suara?

## 1.3 Tujuan

Berdasarkan rumusan masalah didapatkan tujuan sebagai berikut:

1. Membuat perancangan sistem kontrol kipas dan lampu secara implisit menggunakan suara
2. Mengimplementasikan sistem kontrol kipas dan lampu secara implisit dengan menggunakan metode *Text Processing*
3. Melakukan pengujian, dan analisis hasil pengujian terhadap sistem yang telah dibuat

## 1.4 Manfaat

Manfaat yang diperoleh dari rancang bangun sistem kendali kipas dan lampu secara implisit melalui suara dengan menggunakan *Text Processing* adalah:

1. Membuat purwarupa sistem kendali kipas dan lampu pengendalian berbasis *voice command*
2. Menjadi salah satu solusi dalam melakukan kontrol terhadap kipas dan lampu menggunakan suara
3. Mengetahui tingkat efektivitas *speech recognition* sebagai kontrol kipas dan lampu
4. Menjadi salah satu rujukan penelitian yang terkait

## 1.5 Batasan masalah

Batasan masalah yang digunakan dalam penelitian ini agar tujuan penelitian dapat tercapai dan tidak menyimpang yaitu:

1. Pengolahan *speech recognition* menggunakan Android *device*.
2. Sistem yang dibuat diuji di dalam ruangan.
3. Perintah yang digunakan berupa kalimat implisit.
4. Output yang dihasilkan berupa kontrol terhadap lampu dan kipas.

## **1.6 Sistematika pembahasan**

Sistematika pembahasan ini berisi tentang penjelasan secara umum, struktur dan isi dari bab – bab yang ada dalam penelitian ini. Adapun sistematika pembahasan penelitian ini adalah sebagai berikut:

### **BAB I: Pendahuluan**

Pada bab I menjelaskan latar belakang permasalahan, rumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika pembahasan.

### **BAB II: Dasar Teori**

Pada bab II membahas tentang penjelasan teori-teori dasar dan tinjauan pustaka yang menjadi acuan dalam melaksanakan penerapan penelitian.

### **BAB III: Metodologi Penelitian**

Pada bab III ini membahas tentang analisis kebutuhan, perancangan, implementasi, pengujian serta analisis dari sistem, dan pengambilan saran dan kesimpulan.

### **BAB IV: Analisis Kebutuhan**

Pada Bab IV membahas proses implementasi sistem dari penelitian kebutuhan sistem, kebutuhan pengguna, kebutuhan *software*, dan kebutuhan *hardware*.

### **BAB V: Perancangan sistem dan Implementasi**

Pada bab V berisi penjelasan mengenai perancangan dan implementasi sistem, seperti diagram komunikasi sistem dan diagram alur kerja sistem.

### **BAB VI: Pengujian dan Analisis**

Pada bab VI berisi penjelasan hasil proses pengujian sistem, prosedur pengujian, hasil dan analisis dari pengujian yang telah dilakukan.

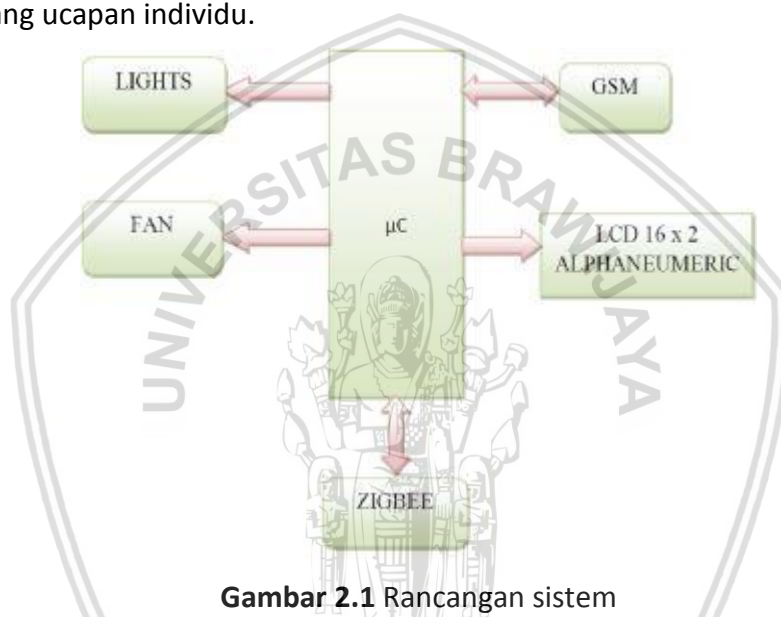
### **BAB VII: Penutup**

Pada bab VII berisi tentang kesimpulan yang didapat oleh peneliti dan saran untuk pengembangan sistem kedepannya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Penelitian yang telah dilakukan sebelumnya yang berjudul “*Voice Controlled Smart Home*” yang dilakukan oleh (Amrutha S,2015) menghasilkan sebuah sistem dengan suara sebagai dasar perintah untuk melakukan kontrol terhadap berbagai peralatan rumah tangga secara otomatis. Penelitian ini menggunakan suara sebagai *input* dengan memanfaatkan HM 2007 *speech recognition chip*. *Chip* ini mengirimkan Perintah suara dalam bentuk bahasa biner yang dimengerti oleh mikrokontroller. Kata-kata tertentu dari pembicara tertentu akan secara otomatis diakui yang didasarkan pada informasi yang termasuk dalam gelombang ucapan individu.



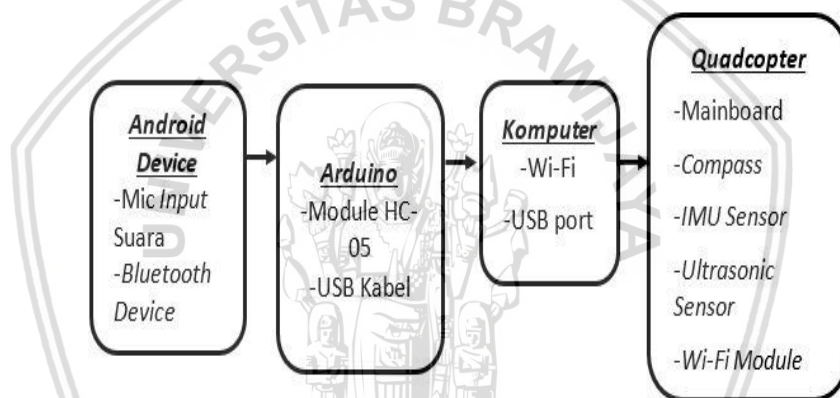
**Gambar 2.1** Rancangan sistem

Penelitian ini menggunakan Zigbee sebagai media untuk sebagai media *Wireless connection*. Setelah *input* suara diterima mikrokontroller akan mengolah data *input* tersebut untuk menghasilkan perintah terhadap kontrol lampu dan kipas. Makalah ini menjelaskan penemuan serta kemajuan teknologi di bidang pengenalan suara dan juga berfokus pada langkah-langkah yang berbeda yang terlibat untuk identifikasi pembicara menggunakan MATLAB Programming. Sistem rumah pintar nirkabel yang dikontrol dengan suara telah disajikan untuk orang tua dan orang dengan kekurangan fisik. Sistem yang diusulkan memiliki dua komponen utama yaitu: sistem pengenalan suara, dan sistem nirkabel. Setiap beban rumah akan memiliki dua perintah ON dan OFF. Otomatisasi 2 beban seperti kipas dan cahaya telah diuji dengan memberikan 4 perintah suara melalui komputer pribadi. Ketika pengguna membuat profilnya sendiri dan mengotomatiskan akurasi pengenalan beban beban lebih dari 90% tercapai. Orang lain yang diizinkan untuk mengotomatiskan beban oleh pengguna dapat menggunakan profil pengguna dan mencapai akurasi pengenalan ucapan dari 75% di komputer pribadi yang sama.



Penelitian rujukan yang kedua berjudul “Sistem kendali navigasi *quadcopter* menggunakan suara melalui *smartphone* dan arduino dengan metode *text processing*” amenerapkan pengolahan suara sebagai kontrol *quadcopter*. *Quadcopter* dikendalikan melalui suara dengan perantara *Android device* menggunakan *speech recognition Google* dan melakukan *text processing* dengan *database SQLite* untuk diproses menjadi gerakan *quadcopter*. Penelitian yang dikerjakan menggunakan *text processing* yang merupakan bagian dari *text mining* yang mana metode ini berfokus pada pengolahan kalimat. Kata hasil pengolahan dimasukkan ke dalam indeks dan diolah dengan mengelompokkan kata inti dan kata imbuhan.

Perintah yang dimasukkan berupa kata seperti : naik, terbang, kiri, kanan, atas, bawah. *Text Processing* mengolah kata – kata *input* dan membedakan antara kata imbuhan dan kata inti. Sehingga *quadcopter* dapat terbang sesuai dengan keinginan *user*. Data suara yang masuk diubah menjadi teks dan kalimat yang masuk diubah menjadi satuan kata – kata.



**Gambar 2.2** Proses kendali quadcopter menggunakan suara

Sumber : (Pallas,2017)

Alur proses kendali quadcopter menggunakan suara dapat dilihat pada Gambar 2.1 awalnya user memberikan masukan suara melalui mic yang ada di *Android Device*. Masukan suara diolah di dalam *Arduino* kemudian akan dikirimkan ke komputer melalui *usb* untuk menjalankan program *Node.js*. Program *Node.js* digunakan untuk mengontrol gerakan *Quadcopter*. Gerakan *Quadcopter* yang dapat dihasilkan mencapai 5 gerakan.

## 2.2 Dasar Teori

Pada bagian ini akan membahas teori – teori yang berhubungan dengan penelitian, *hardware*, *software* yang akan digunakan pada penelitian.

### 2.2.1 Smart Home

*Smart Home* adalah sebuah sistem otomatis yang sangat canggih untuk mengendalikan peralatan - peralatan elektronik untuk memantau sekaligus mengaktifkan keamanan (Bregman D, 2010). Menjelaskan pengertian dan tujuan



dari dibangunnya Smart Home, dimana Smart Home memberikan solusi alternatif dalam efisiensi penggunaan energi listrik. Teknologi mempunyai kemampuan untuk bekerja secara 'pintar' yaitu mampu memonitor, mengontrol, mengantisipasi, memprediksi dan membuat keputusan.



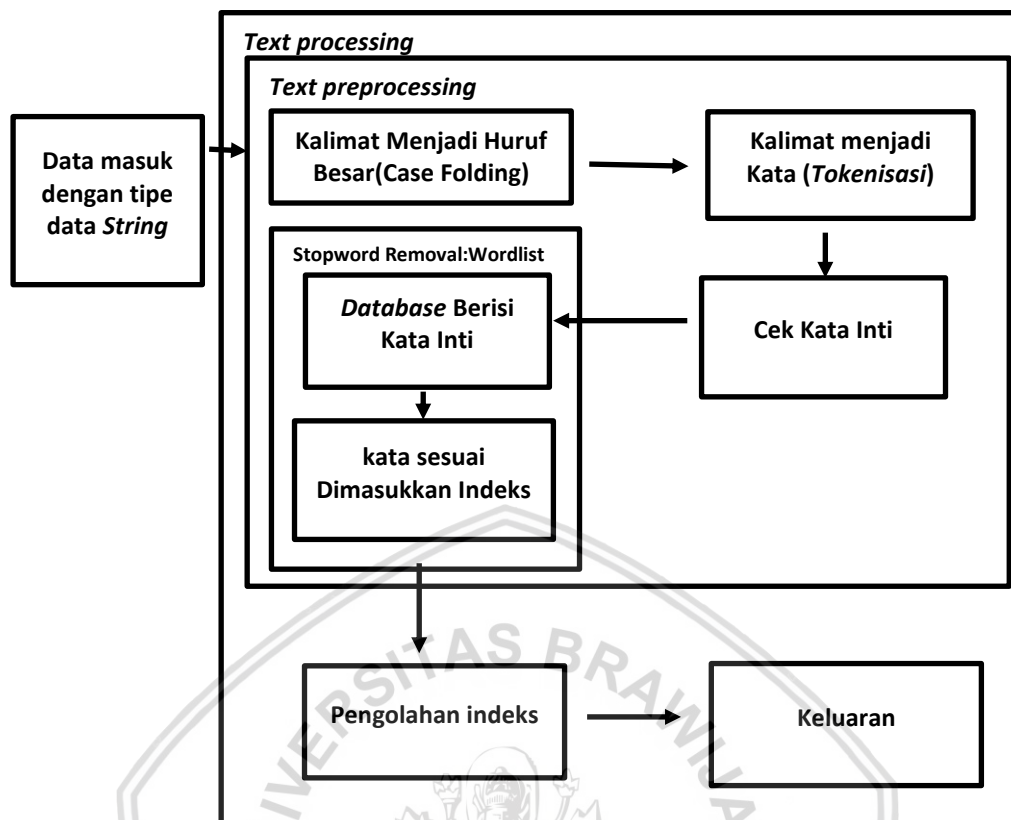
**Gambar 2.3 Smart Home**

Sumber : [www.rumah123.com](http://www.rumah123.com) (2017)

*Smart Home* memiliki potensi untuk meningkatkan kenyamanan, keamanan dan hiburan bagi orang yang memakainya (Meyer et al, 2003). Keuntungan dari teknologi *Smart Home* dapat dipakai untuk seluruh kalangan, seperti orang kekurangan fisik, orang tua, anak-anak, dan lain-lain. Kontrol *Smart Home* sendiri dapat dilakukan dengan berbagai interaksi, seperti suara, gerakan, dan lain – lain. Penulis sendiri akan fokus kepada kontrol menggunakan suara.

### **2.2.2 Text Processing**

*Text processing* merupakan bagian dari *text mining* yang mana metode ini berfokus pada pengolahan kalimat. Kata hasil pengolahan dimasukkan ke dalam indeks dan diolah dengan mengelompokkan kata inti dan imbuhan. Alur metode *text processing* yang digunakan dapat dilihat pada Gambar 2.4.



Gambar 2.4 Alur Metode Text Processing

Sumber : (Pallas,2017)

Masukan dari metode ini berupa kalimat dengan tipe data *string*. Langkah pertama, kalimat yang masuk diubah menjadi huruf besar tahap ini dinamakan *case folding*. Selanjutnya kalimat-kalimat tersebut dipecah menjadi satuan kata, tahap ini dinamakan *tokenisasi*. Tahap pemilihan kata menggunakan *stopword removal* dengan algoritme *wordlist*, tahap ini untuk memilih kata-kata penting dan menyaring kata-kata yang tidak dibutuhkan. Hasil dari tokenisasi ini, dicari di dalam *database* yang berisi kumpulan-kumpulan dari kata inti, jika kata ditemukan di dalam *database* maka data akan disimpan di dalam indeks (Fauzi, 2016). Kumpulan kata indeks yang di dapat dikelompokkan, diambil inti dari kalimat dan diolah menjadi keluaran yang diinginkan.

### 2.2.3 Arduino

Arduino merupakan salah satu vendor penyedia mikrontroler yang banyak digubakan untuk melakukan purwarupa suatu sistem. Arduino memiliki banyak tipe, mulai dari yang paling sederhana, dengan kemampuan dan spesifikasi sederhana, hingga yang menggunakan prosesor dan fungsi yang kompleks. Sebuah mikrokontroler Arduino terdiri dari IC utama yang berfungsi sebagai pengendali utama dari jalannya sistem mikrokontroler tersebut. Pada mikrontroler ini juga dilengkapi dengan pin *input / output* yang jumlahnya berbeda bergantung pada tipe Arduino tersebut. Pin tersebut memiliki fungsi khusus, seperti PWM, pin digital, dan pin analog.

Namun dari semua tipe Arduino, Arduino Mega 2560 merupakan tipe yang paling banyak digunakan. Arduino mega banyak digunakan karena bersifat *open source*, memiliki banyak pin digital dan pin analog yang dapat dimanfaatkan untuk membuat sistem yang lebih kompleks, serta memiliki harga yang terjangkau. Arduino mega mampu mengakomodir kebutuhan komputasi dan menangani lebih dari satu *input* dari sensor maupun perangkat lain serta mampu menghasilkan *output* dalam bentuk aktuator maupun dalam bentuk lain.

Arduino juga menyediakan aplikasi untuk memasukkan kode program yaitu Arduino IDE. Setelah kode program selesai dibuat, maka *board* Arduino mega dapat dihubungkan ke komputer yang memiliki aplikasi tersebut untuk melakukan *upload* program yang telah dibuat ke board Arduino Mega. *Upload* program ini dapat dilakukan berulang, sehingga apabila terjadi kesalahan, program dapat di *upload* ulang. Untuk melakukan *upload* program, membutuhkan bantuan kabel yang menghubungkan Arduino dengan komputer.

Arduino dapat dihubungkan dengan sensor maupun aktuator yang mendukung (Geddes, 2016).



**Gambar 2.5** Arduino Mega 2560

Sumber : (Arduino, 2017)

**Tabel 2.1 Spesifikasi Arduino Mega 2560**

Chip Mikrokontroler	ATmega 2560
Tegangan Operasi	5V
Rekomendasi Tegangan <i>Input</i>	7V - 12V
Tegangan <i>Input</i> Maksimal	6V 20V
Pin I/O Digital	54 pin, 15 diantaranya PWM
Pin <i>input</i> analog	16 pin
Arus DC pin I/O	20mA
Arus DC pin 3.3V	50mA
Flash Memori	256KB, 8KB untuk <i>bootloader</i>
SRAM	8KB
EEPROM	4KB
<i>Clock Speed</i>	16Mhz
Dimensi	101.52 mm x 53.3mm
Berat	37gr

#### 2.2.4 HC-05

Adalah sebuah modul *Bluetooth* SPP (*Serial Port Protocol*) yang banyak digunakan untuk komunikasi tanpa kabel, prinsip kerjanya yaitu mengkonversi *port* serial ke *Bluetooth*. HC-05 memiliki dua mode konfigurasi, yaitu *AT mode* dan *Communication mode*. *AT mode* berfungsi untuk melakukan pengaturan konfigurasi dari HC-05, Sedangkan *Communication mode* berfungsi untuk melakukan komunikasi. Pada *Bluetooth* dengan piranti lain. Jarak sinyal dari HC-05 adalah 30 meter, dengan kondisi tanpa halangan (TerryKing, 2011).

**Gambar 2.6 HC-05**

Sumber: (TerryKing, 2011)

Mode komunikasi yang digunakan berupa mode *Slave*, mode ini memungkinkan Android *device* untuk terhubung dengan modul ini. Digunakan tegangan 5V supaya memiliki performa yang maksimal dalam penerimaan data.

### 2.2.5 Relay 2 Channel Output

*Relay module* adalah sebuah saklar listrik yang dapat menghidupkan atau mematikan sebuah arus listrik dengan menggunakan voltase dan atau arus yang jauh lebih tinggi daripada yang dapat ditangani oleh mikrokontroller. *Relay* juga dapat digunakan untuk menghubungkan antara arus DC dan arus AC.



**Gambar 2.7** Relay 5V DC 2 Channel Output

Sumber: ([www.uetronics.com](http://www.uetronics.com))

*Relay* yang dipakai untuk penelitian ini, spesifikasinya adalah sebagai berikut:

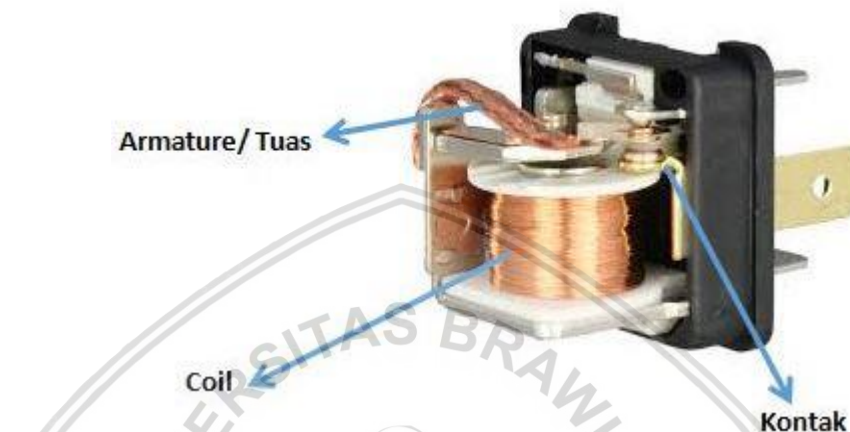
Interface *input* / pemicu:

- DC+ : power +5 V DC
- DC- : power -5 V DC
- IN1 : sinyal *low* atau *high* pada *channel* 1
- IN2 : sinyal *low* atau *high* pada *channel* 2

Interface *output relay*:

- NO 1 : *normally open channel* 1 ( *relay* tidak hidup sampai ada sinyal baru hidup )
- COM 1 : *common interface channel* 1
- NC 1 : *normally else channel* 1 ( *relay* hidup sampai ada sinyal baru yang mati )

- NO 2 : *normally open channel 2* (relay tidak hidup sampai ada sinyal baru hidup)
- COM 2 : *common interface channel 2*
- NC 2 : *normally colse channel 1* ( relay hidup sampai ada sinyal baru yang mati)



**Gambar 2.8** Bagian – bagian dari *relay*

Sumber: (<http://panduanteknisi.com>)

Secara umum, bagian – bagian dalam *relay* terdiri dari:

- *Electromagnet*
- *Armature* ( tuas mekanik )
- Kontak switch
- Pegas / per

Ketika kumparan *electromagnet* yang ada didalamnya terdapat sebuah logam *ferromagnetis* dialiri arus listrik, akan timbul medan magnet dan akan menarik tuas mekanik sehingga mengubah posisi kontak *switch* yang ada. Dari sebelumnya NC menjadi NO.

### 2.2.6 *Speech recognition Android Device*

*Speech recognition* adalah proses indentifikasi suara berdasarkan kata yang diucapkan dengan melakukan konversi sebuah sinyal akustik. Berbeda dengan *Sound Recognition*, *Speech Recognition* lebih fokus kepada apa yang diucapkan daripada siapa yang mengucapakan.

Terdapat empat langkah utama dalam sistem pengenalan suara:

- Penerimaan data masukan.
- *Ekstrasi*, yaitu menyimpan data masukan sekaligus pembuatan *database* untuk *template*.



- Pembadingan / pencocokan, yaitu tahap pencocokan data baru dengan data suara (pencocokan tata bahasa) pada *template*.
- Validasi identitas pengguna.

Langkah pertama dari pengolahan *speech recognition* yaitu sinyal suara masuk ke dalam suatu platform, lalu dilakukan ekstraksi fitur yaitu pengambilan nilai suara berupa frekuensi agar nantinya bisa diolah lebih lanjut. Langkah selanjutnya yaitu mencocokkan hasil ekstraksi fitur dengan *database* yang ada, dengan hasil akhir didapatkan berupa deteksi kata yang telah diucapkan.

Dalam penggunaanya *speech recognition* dapat diolah menggunakan berbagai macam *platform*, seperti Matlab, LabView, Android *device*, Arduino dan masih banyak *platform* lainnya.

### 2.2.7 Penghitungan Persentase Akurasi

Nilai Persentase Akurasi suatu sistem dapat dihitung menggunakan rumus 2.1

$$\text{Nilai Persentase Akurasi} = \frac{\text{Jumlah nilai Benar}}{\text{Jumlah nilai keseluruhan}} \times 100\% \quad (2.1)$$

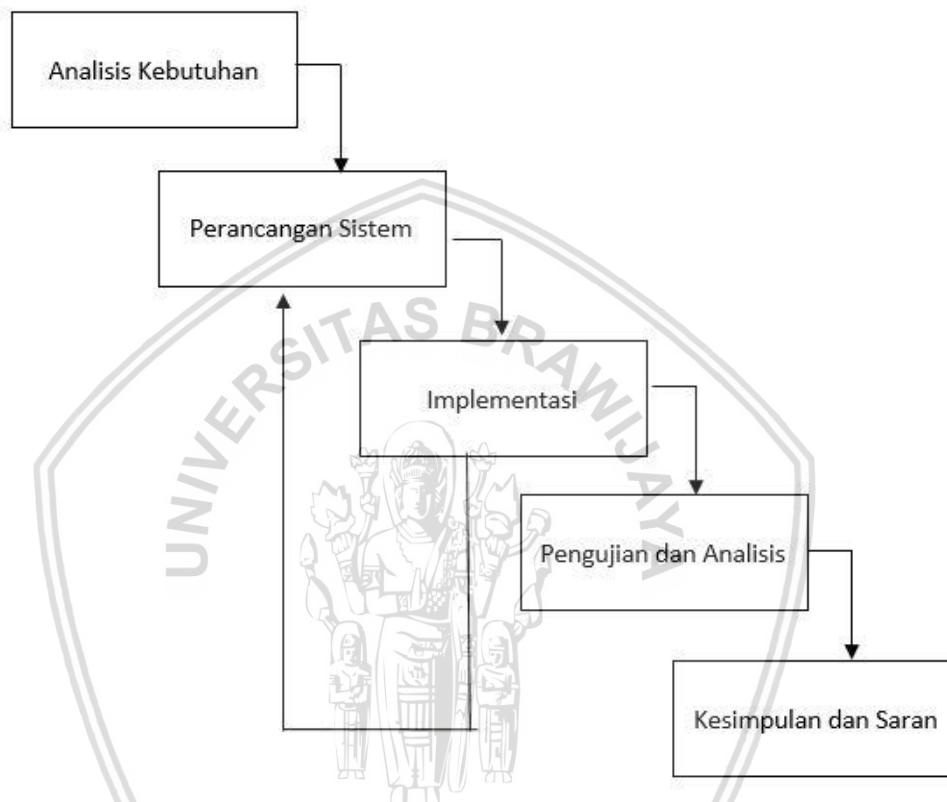
Nilai Persentase Akurasi diperoleh dari Nilai data benar pada suatu sistem dibagi dengan nilai data keseluruhan.



## BAB 3 METODOLOGI

### 3.1 Metodologi Penelitian

Untuk melaksanakan penelitian, dan agar penelitian dapat berjalan dengan baik dan dapat mencapai tujuan yang diinginkan, maka diperlukan perencanaan urutan kegiatan dari awal dalam diagram sebagai berikut:



**Gambar 3.1** Alur Metodologi Penelitian

Pada gambar 3.1 merupakan metodologi penelitian yang digunakan, pada bab analisis kebutuhan akan dibahas kebutuhan yang digunakan untuk penelitian ini. Pada bab perancangan sistem dan implementasi akan dibahas mengenai bagaimana merancang sistem yang dibuat lalu diimplementasi, namun apabila implementasi tidak sesuai dengan yang diharapkan maka akan dirancang kembali. Jika sesuai dengan harapan, maka akan dilanjut ke bab pengujian dan analisis. Pada bab pengujian dan analisis hasil yang telah dirancang akan diuji dan dianalisis hasil pengujiannya lalu ditarik kesimpulan dan pengambilan saran.

### 3.2 Analisis kebutuhan

Analisa kebutuhan merupakan penentuan perangkat pendukung untuk mengimplementasikan sistem baik perangkat keras, perangkat lunak, kebutuhan fungsional, dan kebutuhan nonfungsional dari sistem dengan spesifikasi yang

sesuai dengan kebutuhan sistem yang akan dibangun yaitu sistem kendali kipas dan lampu secara implisit dengan metode *Text Processing*.

Perangkat keras yang akan digunakan dalam sistem ini adalah:

1. *Arduino mega*, sebagai pengendali utama sistem.
2. *Android*, sebagai *input* suara.
3. Lampu, sebagai *output* cahaya
4. Kipas, sebagai *output* suhu

Perangkat lunak yang akan digunakan dalam sistem ini adalah:

1. *Arduino IDE*
2. *Android Studio*

Kebutuhan fungsional sistem yaitu:

1. Sistem dapat mengenali perintah yang dimasukan manusia menjadi sebuah kontrol terhadap kipas dan lampu
2. Sistem harus dapat menentukan menghidupkan atau mematikan perangkat kerasnya berdasarkan pengolahan data yang dilakukan
3. Kipas mewakili kontrol suhu ruangan dan Lampu mewakili kontrol cahaya terhadap ruangan

Kebutuhan non fungsional dari sistem yaitu :

1. Baterai yang digunakan pada android yang digunakan pada percobaan sistem

### 3.3 Perancangan Sistem

Perancangan sistem merupakan tahapan penelitian sistem didesain agar dapat memenuhi kebutuhan fungsional dan bekerja sesuai dengan yang diharapkan peneliti. Teori dan literatur yang mendukung serta pengetahuan yang didapatkan selama perkuliahan menjadi dasar dan panduan dalam implementasi kontrol lampu dan kipas secara implisit menggunakan suara dengan metode *text processing*.

### 3.4 Implementasi Sistem

Setelah tahapan perencanaan dan perancangan selesai dilakukan maka dilanjutkan dalam tahap realisasi dari semua perancangan dan perencanaan yang telah dilakukan sebelumnya. Dalam tahapan implementasi ini terdapat beberapa fungsi dari sistem yang harus dapat berajalan, yaitu:

1. Perangkat keras yang digunakan harus dapat berjalan dan berfungsi sesuai dengan spesifikasi yang terdapat pada *datasheet* produk tersebut.
2. Perangkat keras harus dapat mengontrol kipas angin dan lampu dengan menggunakan suara sesuai dengan metode yang digunakan.

### 3.5 Pengujian dan Analisis Sistem

Pengujian dan analisis dilakukan untuk mengetahui kesesuaian dan performa dari sistem. Kesesuaian merupakan kemampuan sistem menghasilkan keluaran sesuai dengan *input* yang diberikan dan juga proses yang dilakukan oleh sistem. Performa sistem merupakan kemampuan sistem untuk melakukan proses pengolahan data *input* sehingga menghasilkan *output* dengan baik dan benar serta waktu proses yang singkat. Berikut adalah parameter yang digunakan dalam pengujian sistem:

1. Pengujian fungsional dari sistem. Hal ini ditujukan untuk mengetahui apakah sistem sudah berjalan sesuai dengan keinginan peneliti. Tolok ukur dalam hal ini dapat dilihat melalui *console log* pada Android Studio dan serial monitor pada Arduino.
2. Pengujian masukan kalimat dan pengolahan teks sistem. Pengujian ini digunakan untuk melihat akurasi dari pengolahan suara dan pengolahan teks, dengan menggunakan individu yang berbeda.
3. Pengujian efektivitas sistem dengan individu yang berbeda-beda. Individu yang berbeda digunakan untuk menguji sistem, apakah sistem mampu mengolah masukan dan menghasilkan keluaran sesuai dengan harapan *user*.
4. Pengujian waktu yang diperlukan aplikasi untuk mengolah perintah dari *user*. Tahap ini untuk menguji *delay* dari penggunaan perintah suara jika digunakan untuk kontrol sistem.

### 3.6 Penarikan kesimpulan dan saran

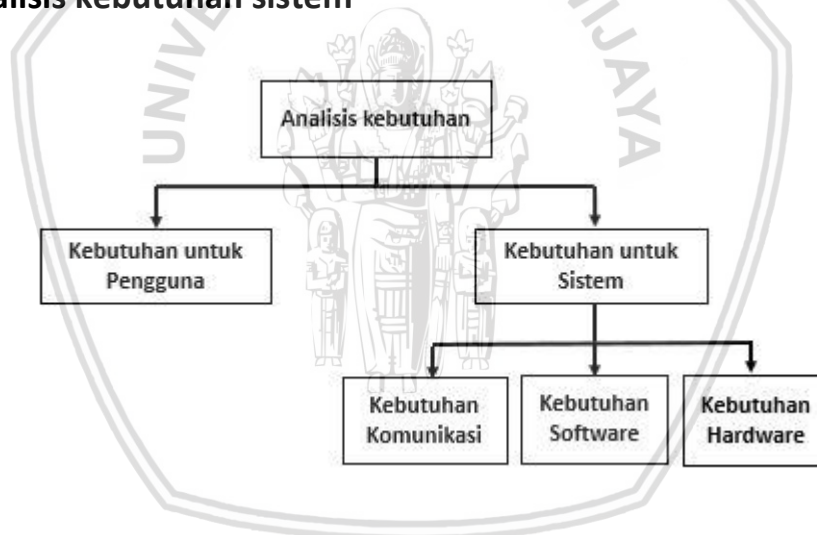
Pengambilan kesimpulan dilakukan berdasarkan pengujian dan analisa yang telah dilakukan terhadap kesesuaian dan performa dari sistem. Kesimpulan yang diambil juga harus dapat menjawab rumusan permasalahan yang terdapat pada bab 1. Dari keseluruhan penelitian yang telah dilakukan ini, maka diharapkan dapat diberikan saran untuk pengembangan sistem di masa yang akan datang.

## BAB 4 ANALISIS KEBUTUHAN

### 4.1 Gambaran umum sistem

Gambaran umum pada sistem yang dibuat, lampu dan kipas mampu menyala sesuai perintah implisit yang dimasukkan oleh *user* melalui *Android device*. Sistem yang digunakan berupa *Android device* yang digunakan untuk *input* suara sekaligus tempat untuk terjadinya *text preprocessing*, tahap ini dimulai dari *case folding* yaitu mengubah semua huruf *input* menjadi huruf besar. Kemudian pemecahan kalimat menjadi kata, lalu setiap kata tersebut akan dicocokkan di dalam database yang telah ada dengan menggunakan langkah *stopword removal*, *stopword removal* ini menggunakan algoritme *wordlist*. Jika ada kata sesuai dengan salah satu isi database yang ada, maka akan dikirim ke *arduino* melalui koneksi *bluetooth*. Koneksi *bluetooth* pada *arduino* ini menggunakan modul *HC-05* sebagai receiver yang berfungsi untuk menerima data dari *Android* untuk dilakukan *text processing* yang nantinya akan dikirim ke *L298N* dan atau *Relay* untuk mengirimkan perintah untuk menghidupkan kipas dan lampu.

### 4.2 Analisis kebutuhan sistem



**Gambar 4.1** Analisis kebutuhan sistem

Pada bagian ini akan dibahas lebih lanjut mengenai kebutuhan-kebutuhan yang digunakan untuk sistem ini, seperti kebutuhan pengguna, kebutuhan *hardware*, kebutuhan *software*, dan yang terakhir kebutuhan komunikasi. Mengacu pada Gambar 4.1 semua komponen dalam analisis kebutuhan saling berkaitan satu sama lain, hal ini dikarenakan sistem ini membutuhkan semua komponen pada Gambar 4.1 agar berjalan sesuai dengan keinginan.

#### 4.2.1 Kebutuhan pengguna

Kebutuhan pengguna menggambarkan hal-hal yang dilakukan oleh pengguna agar sistem ini berjalan sesuai dengan keinginan penggunanya, yaitu:

1. Pengguna membutuhkan aplikasi Android yang telah dibuat oleh peneliti dengan nama *Smart Home* sebagai masukan suara dan koneksi *bluetooth* untuk inisialisasi bahwa sistem dimulai.
2. Pengguna dapat melihat history perintah yang telah dimasukkan pada aplikasi Android yang telah dibuat oleh peneliti.

#### 4.2.2 Kebutuhan sistem

Kebutuhan sistem meliputi hal – hal yang dibutuhkan agar sistem dapat berjalan sesuai dengan keinginan peneliti yang terdiri dari kebutuhan *hardware*, *software* dan komunikasi.

##### 4.2.2.1 Kebutuhan Hardware

Kebutuhan *hardware* akan dijelaskan lebih lanjut, yaitu:

1. Android device

*Smartphone* atau bisa disebut Android device merupakan kebutuhan *hardware* yang berfungsi sebagai tempat pengolahan serta masukan suara dari *user*. Android device yang digunakan harus memiliki *Bluetooth device*, tidak ada syarat minimal spesifikasi Android device yang akan digunakan. Namun dalam penelitian ini spesifikasi Android device yang digunakan dapat dilihat pada Tabel 4.1

**Tabel 4.1** Spesifikasi Android Device yang Digunakan

Nama	Spesifikasi
Merek	Asus Zenfone Max Pro (M1) ZB601KL
API	Api-22
Sistem Operasi	Android 8.1 Oreo
Memori	4 GB
Internal Memori	64 GB
GPU	Adreno 509
Chipset	Qualcomm SDM636 Snapdragon 636
Bluetooth	4.2, A2DP, LE

Pada Tabel 4.1 Android *device* yang digunakan Asus Zenfone Max Pro (M1) ZB601KL, memiliki sistem operasi Android 8.1 *Oreo*, dengan RAM 4 GB dan *Internal* memori : 64 GB. *Chipset* pada Zenfone Max Pro (M1) ZB601KL digunakan Qualcomm SDM636 Snapdragon 636 *Processor octa-core 1.8 Hz Kryo*.

*Hardware* yang dibutuhkan pada android *device* diantaranya:

1. *Mic* sebagai media untuk masukan suara dari *user*.
2. *Bluetooth device* digunakan untuk komunikasi antara Android dengan Arduino.

## 2. Arduino

Arduino merupakan *hardware* yang bersifat *open source* dan *software*nya mudah dikembangkan. Rician spesifikasi Arduino dapat dilihat pada Tabel 4.2.

**Tabel 4.2** Spesifikasi Arduino

Komponen	Arduino Mega	Arduino Uno	Arduino Nano
<i>Chip Mikrokontroler</i>	ATmega2560	ATmega328P	ATmega328P
<i>SRAM</i>	8KB	2KB	2KB
<i>EEPROM</i>	4KB	1KB	1KB
<i>Memori Flash</i>	256 KB	32 KB	32 KB

pada penelitian ini menggunakan Arduino Mega karena memiliki spesifikasi lebih bagus daripada jenis Arduino lainnya. *Hardware* yang dihubungkan dengan Arduino diantaranya:

- a. *Module Bluetooth* HC-05 dihubungkan dengan Arduino berfungsi sebagai penerima data dari Android *device*.
- b. *Kabel USB* berfungsi untuk menghubungkan Arduino dengan komputer sekaligus memberikan daya untuk menyalakan Arduino.
- c. Kipas 5v
- d. Lampu

## 3. Komputer

Komputer sebagai media untuk pembuatan aplikasi kebutuhan pengguna. Spesifikasi komputer yang digunakan untuk penelitian ini dapat dilihat pada Tabel 4.3



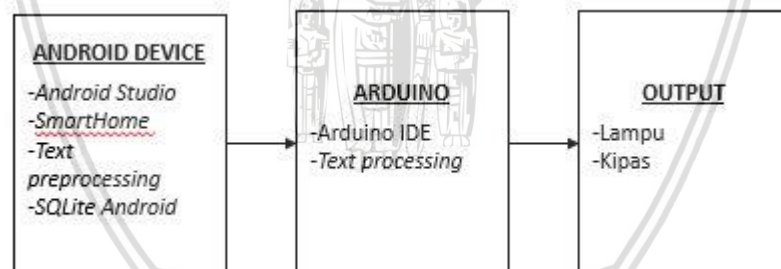
**Tabel 4.3** Spesifikasi Komputer yang Digunakan

Nama	Spesifikasi
Merk	Lenovo
BIOS	InsydeH2O V1.04
Sistem operasi	Windows 7 Ultimate-64 Bit
Ram	4Gb
Port USB	USB 2.0 atau USB 3.0
Prosesor	Intel(R) Pentium(R) CPU B960

Spesifikasi komputer yang digunakan menggunakan Sistem operasi windows 7 ultimate-64 Bit dengan ram 4GB dan prosesor Intel Intel(R) Pentium(R) CPU B960. *Usb Port* yang akan digunakan bisa menggunakan USB 2.0 maupun USB 3.0 untuk menghubungkan komputer dengan Arduino.

#### 4.2.2.2 Kebutuhan *software*

Pada bagian ini akan dijelaskan *software* apa saja yang dibutuhkan untuk melakukan penelitian, dapat dilihat pada Gambar 4.2 skema kebutuhan *software*.

**Gambar 4.2** Kebutuhan *Software*

Kebutuhan *Software* Gambar 4.2 akan dijelaskan lebih lanjut, yaitu:

1. *Android device*

Android memiliki peran penting dalam penelitian ini, pada membutuhkan beberapa *software* untuk pengembangannya di antaranya:

- a. *Android Studio* Merupakan *platform* untuk mengembangkan aplikasi berbasis *mobile* dengan sistem operasi Android. Android studio merupakan *software* yang diperkenalkan pada tahun 2013 oleh



Google. Pada penelitian ini Android studio berfungsi sebagai tempat pembuatan aplikasi *Smart Home*.

- b. *Smart Home* adalah aplikasi yang dibuat oleh peneliti akan digunakan untuk masukan perintah suara dari *user*, nantinya hasil dari masukan suara *user* akan diolah lebih lanjut di aplikasi ini.
- c. *Text preprocessing* Merupakan tahap di mana hasil dari masukan suara berupa kalimat dengan tipe data *String* diubah menjadi huruf besar lalu dipecah menjadi beberapa potongan kata. Lalu kata tersebut akan dicocokkan dengan kumpulan kata yang ada pada *database* dengan algoritme *stopword removal wordlist*. Jika ada kata yang sesuai dengan yang ada pada *database*, maka tabel kolom kedua yaitu tabel *command* sebagai perintah kontrol akan dikirim ke Arduino.
- d. SQLite merupakan *database* yang terdapat pada Android, dapat diolah dengan mudah melalui Android studio dengan *query* dasar *database* seperti *Insert, Update, Delete, Select*.

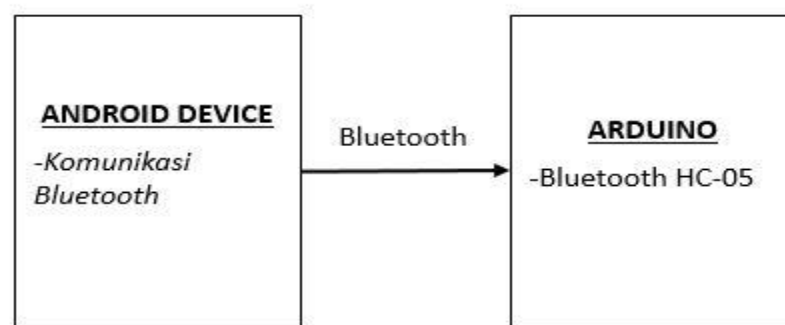
## 2. Arduino

Dalam *Arduino* ada beberapa kebutuhan untk *software* dan penggunaan *library* di antaranya:

- a. *Arduino IDE* merupakan platform untuk mengembangkan kode untuk *Arduino* selain itu, meng-*upload* code algoritme yang kita inginkan ke dalam *Arduino*.
- b. *Text processing* Merupakan tahap data yang diterima dari Android diolah oleh *Arduino*, tiap-tiap kata masukan yang berbeda akan menghasilkan keluaran berupa perintah untuk menghidupkan dan atau mematikan lampu serta kipas.
- c. *Software Serial* merupakan *library* yang tersedia pada *Arduino*. *Library* ini digunakan untuk menghubungkan *Bluetooth* Modul HC-05 dengan *Arduino*.

### 4.2.2.3 Kebutuhan komunikasi

Pada bagian ini akan dijelaskan komunikasi apa saja yang dibutuhkan untuk melakukan penelitian ini, skema kebutuhan *software* dapat dilihat pada Gambar 4.3.



**Gambar 4.3** Kebutuhan Komunikasi

Kebutuhan Komunikasi Gambar 4.3 akan dijelaskan lebih lanjut, yaitu:

1. Untuk menghubungkan *Android device* dengan *Arduino* dibutuhkan komunikasi:
  - a. Komunikasi yang digunakan pada *Android* dan *Arduino* melalui *Bluetooth*, Data hasil pengolahan *Text Preprocessing* dikirim ke *Arduino* melalui *Bluetooth*. *Bluetooth* di *Android* berfungsi sebagai Master, sedangkan pada *Arduino* berfungsi sebagai *slave*, sehingga kedua *device* ini dapat saling terhubung satu sama lain.
2. Di dalam *Arduino* membutuhkan 2 kebutuhan komunikasi yaitu:
  - a. Pada *Arduino* menggunakan HC-05 sebagai modul untuk komunikasi antara *Bluetooth* dengan *Android device*, status *Bluetooth* pada Modul ini adalah berupa *Slave*. *Arduino* sendiri berfungsi untuk menerima hasil *Text preprocessing* untuk nantinya akan dilakukan *Text Processing* melalui komunikasi *Bluetooth*.
  - b. Penggunaan Serial komunikasi digunakan untuk menghubungkan *Arduino* dan *Bluetooth module* HC-05 dengan memanfaatkan *Library SoftwareSerial*. Setelah melakukan *Text Preprocessing* maka hasilnya akan dikirim ke *Arduino* melalui komunikasi Serial.

### 4.3 Kebutuhan fungsional

Kebutuhan Fungsional digunakan untuk memberikan rincian kebutuhan-kebutuhan pada penelitian ini agar sesuai dengan harapan peneliti di antaranya:

- a. Sistem dapat menerima *input* suara yang dimasukkan oleh user melalui perantara android. Pada fungsi ini sistem dapat mengenali *input* suara yang dimasukkan oleh *user* menjadi suatu kalimat dengan menggunakan fitur yang telah ada pada *Android device*, yaitu *Google Speech Recognition*.

- b. Sistem berjalan dengan lancar melalui koneksi *Bluetooth*. Pada fungsi ini, penggunaan komunikasi menggunakan *Bluetooth*, yang mana *Bluetooth* untuk menghubungkan Android *device* dengan Arduino. *Hardware* yang digunakan untuk koneksi *Bluetooth* adalah HC-05.
- c. *Database* yang dipakai akan digunakan untuk mencari kata inti dari kalimat *input* yang dimasukkan oleh *user*. Sistem *database* yang digunakan dalam skripsi ini terdapat pada Android Studio yaitu SQLite. *Database* ini akan mencari kata masukan dari *user* secara akurat, jika masukan sesuai dengan salah satu tabel maka akan dikirim ke tabel kedua yang merupakan tabel *command*. Jika tidak ada yang cocok maka akan dikirimkan angka 0.
- d. Sistem dapat menghidupkan dan mematikan kipas dan lampu, kipas dan lampu disini berfungsi sebagai *output*. *Output* ini sesuai dengan maksud *input* suara yang dimasukkan oleh user melalui perantara *Android*. Dalam sistem ini *Outputnya* ada empat jenis, yaitu : Kipas hidup, kipas mati, lampu hidup, lampu mati.

#### 4.4 Kebutuhan non fungsional

Kebutuhan *non* fungsional merupakan kebutuhan yang menjelaskan mengenai apa saja yang menjadi batasan terhadap kebutuhan perancangan sistem. Adapun kebutuhan *non* fungsional dari sistem ini adalah sebagai berikut.

##### 4.4.1 Karakteristik yang berbeda tiap individu

Sistem ini dibuat untuk *user* secara umum dengan harapan akan mempermudah penggunaan kontrol terhadap lampu dan kipas. Dalam sistem ini *user* yang berbeda mampu mengontrol lampu dan kipas secara implisit.

##### 4.4.2 Lingkungan pengoperasian sistem

Pada dasarnya lingkungan pengoperasian sistem ini dapat dilakukan pada lingkungan didalam ruangan tertutup.

##### 4.4.3 Asumsi ketergantungan tiap komponen

Tiap komponen yang digunakan memiliki ketergantungan satu sama lain agar memiliki hasil yang sesuai dengan harapan.

- a. Aplikasi Android membutuhkan koneksi *Bluetooth* HC-05 untuk terhubung ke Arduino sebagai inisialisasi bahwa program dimulai. Di dalamnya terdapat tombol "mic" untuk memulai memberikan perintah suara.
- b. Menggunakan *module Bluetooth* HC-05 yang mana Pin TX dihubungkan dengan pin 10 pada Arduino, Pin RX dihubungkan dengan pin 11 pada Arduino, Pin VCC dan *Ground* dihubungkan ke Pin 5V dan *Ground* pada Arduino.
- c. Menggunakan *Relay* yang berfungsi sebagai saklar untuk *output* dari sistem yang mana *Relay* dihubungkan pada pin 2 dan pin 3.

#### 4.4.4 Pengambilan data

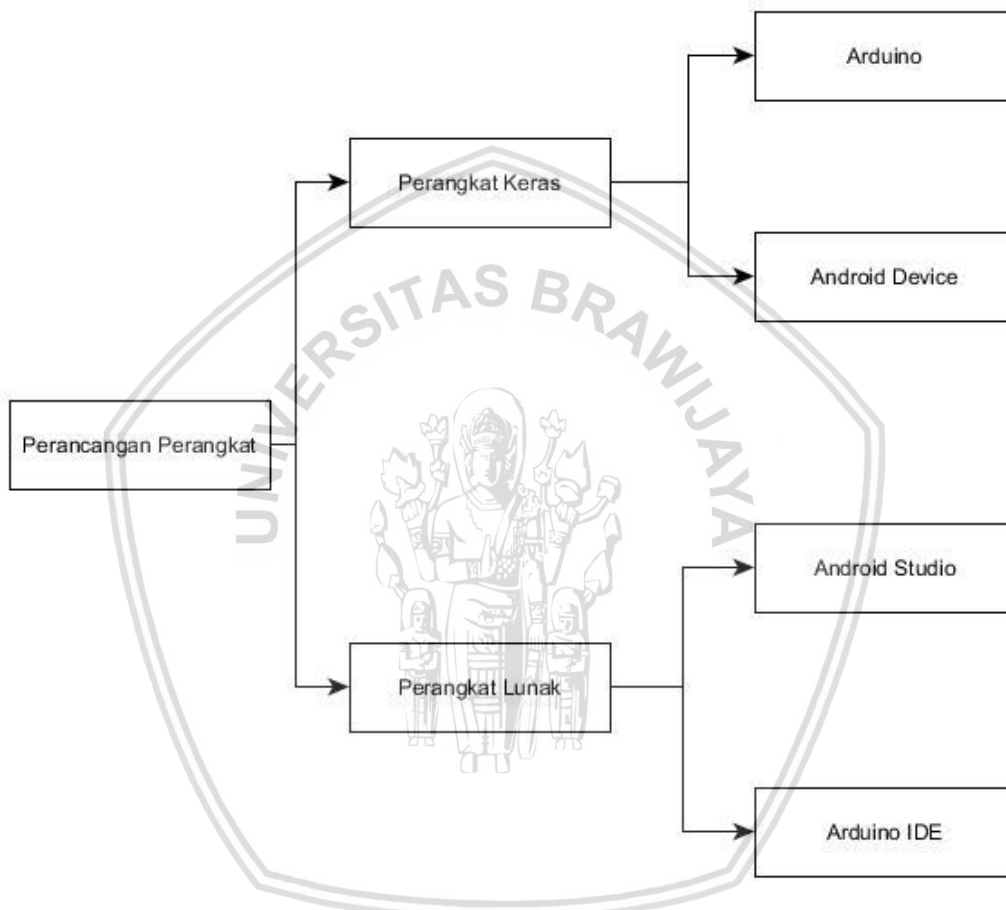
Pengumpulan data dilakukan dengan menggunakan *user* yang berbeda, apakah sistem dapat mengerti *input* suara yang dilakukan *user*. Pengumpulan data yang kedua berupa data *output* yang dihasilkan oleh sistem apakah sistem dapat menghasilkan *output* yang sesuai dengan keinginan *user* tersebut.



## BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM

### 5.1 Perancangan sistem

Perancangan sistem ini penulis bagi menjadi tiga bagian, yaitu perancangan komunikasi antar sistem, perancangan program pada *Android Device*, dan perancangan program pada *Arduino*. Alur Perancangan sistem dapat dilihat pada Gambar 5.1.

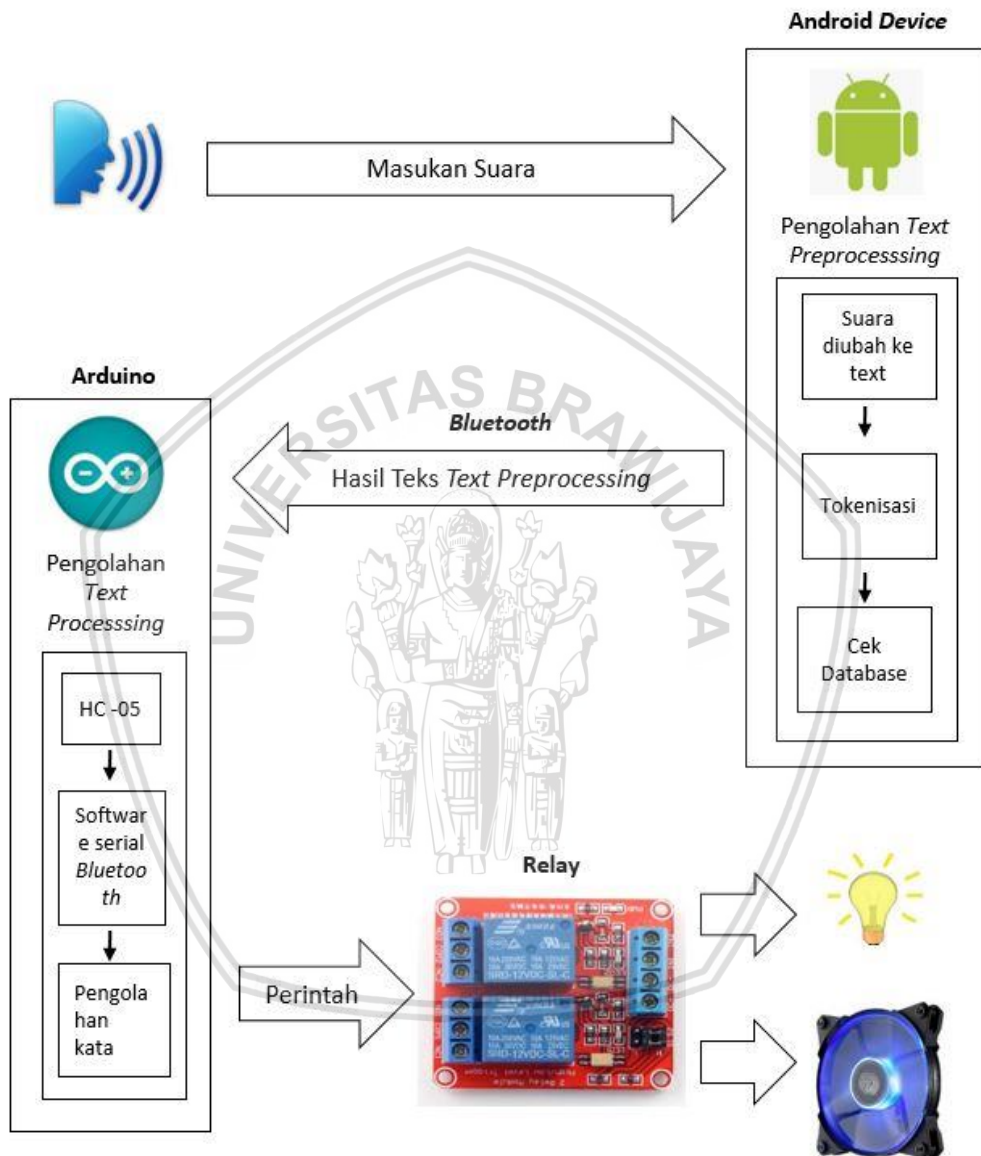


**Gambar 5.1** Alur Perancangan Sistem

Pada Gambar 5.1 menjelaskan secara garis besar alur perancangan sistem, penelitian diawali dengan perancangan perangkat keras / *Hardware* terlebih dahulu, bagaimana setiap komponen yang digunakan dapat terhubung dengan baik. Selanjutnya perancangan program pada *Android* yang digunakan peneliti, program tersebut digunakan sebagai inisialisasi dimulainya sistem. Tahap terakhir yaitu perancangan program pada *arduino*, pada tahap ini *arduino* berfungsi sebagai tempat pengolahan data untuk dijadikan *output* pada sistem, berupa lampu dan kipas.

### 5.1.1 Perancangan Komunikasi antar Sistem

Perancangan komunikasi antar sistem merupakan penjelasan tentang bagaimana setiap komponen yang akan digunakan dalam sistem terhubung. Alur komunikasi program dan proses *text processing* yang digunakan pada penelitian dapat dilihat pada Gambar 5.2.



Gambar 5.2 Perancangan Komunikasi Program

1. Sistem mulai bekerja ketika *user* memberikan *input* suara ke Android *device*
2. *Input* suara diubah menjadi teks dengan memanfaatkan *speech recognition* yang ada pada Android *device*. Hasil pengolahan ini akan menghasilkan data tipe *string*.



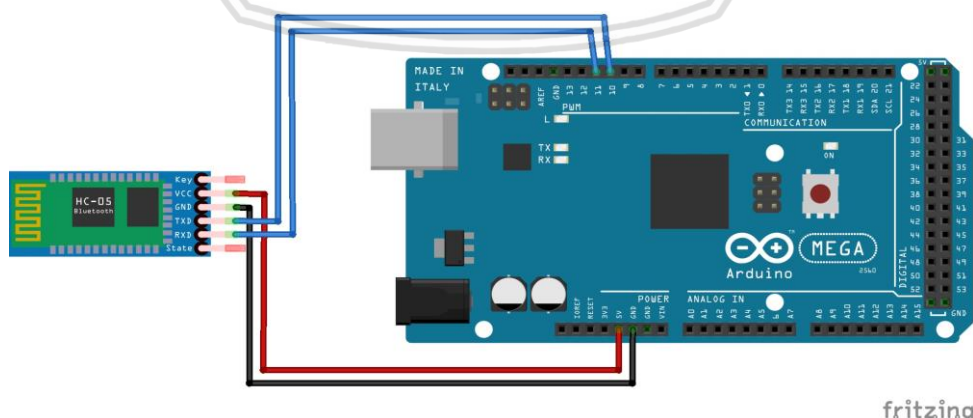
3. Data *string* ini kemudian diolah dengan proses *Text Preprocessing*, yang terdiri dari tokenisasi dan pengecekan *database*.
4. Proses tokenisasi terdapat proses *case folding*, yaitu mengubah semua huruf *inputan* tadi menjadi huruf besar semua. Kemudian dipecah per kata.
5. Setiap kata tersebut akan dicek di dalam *database* dengan algoritme *stopword removal wordlist*.
6. Data hasil pengolahan *text preprocessing* ini akan dikirim ke Arduino melalui koneksi *bluetooth*.
7. Arduino membutuhkan HC-05 untuk komunikasi *bluetooth*.
8. Arduino membutuhkan *software serial* untuk memulai *Bluetooth module* HC-05
9. Data yang dikirim dari Android akan dilakukan pengolahan kata untuk menjadi *output* yang diinginkan.
10. Hasil pengolahan kata dari Arduino akan dimasukkan ke dalam *array*, untuk diolah menjadi *output*.

### 5.1.2 Perancangan rangkaian pada Arduino

Perancangan sistem Arduino pada ini meliputi dua hal, yaitu perancangan pada bagian *hardware* dan perancangan program / *software*. Pada perancangan *hardware* sistem ini terdiri dari Arduino Mega 2650 yang berfungsi sebagai mikrokontroler, HC-05 sebagai *Bluetooth module*, *Relay 2 Channel Output* sebagai penghubung arus DC dan arus AC.

#### 5.1.2.1 Perancangan rangkaian HC-05

HC-05 berfungsi sebagai *Bluetooth module* untuk menghubungkan antara Android *device* dengan Arduino. *Input* suara yang telah diolah pada Android *device* hasilnya tersebut akan dikirim ke Arduino melalui komunikasi *Bluetooth*. Rangkaian penempatan HC-05 ke Arduino dapat dilihat pada gambar 5.3.



**Gambar 5.3** Rangkaian HC-05 ke Arduino Mega 2650

Untuk menghubungkan Arduino Mega 2650 dengan *module Bluetooth* HC-05 menggunakan komunikasi serial, dengan memanfaatkan *library software* serial



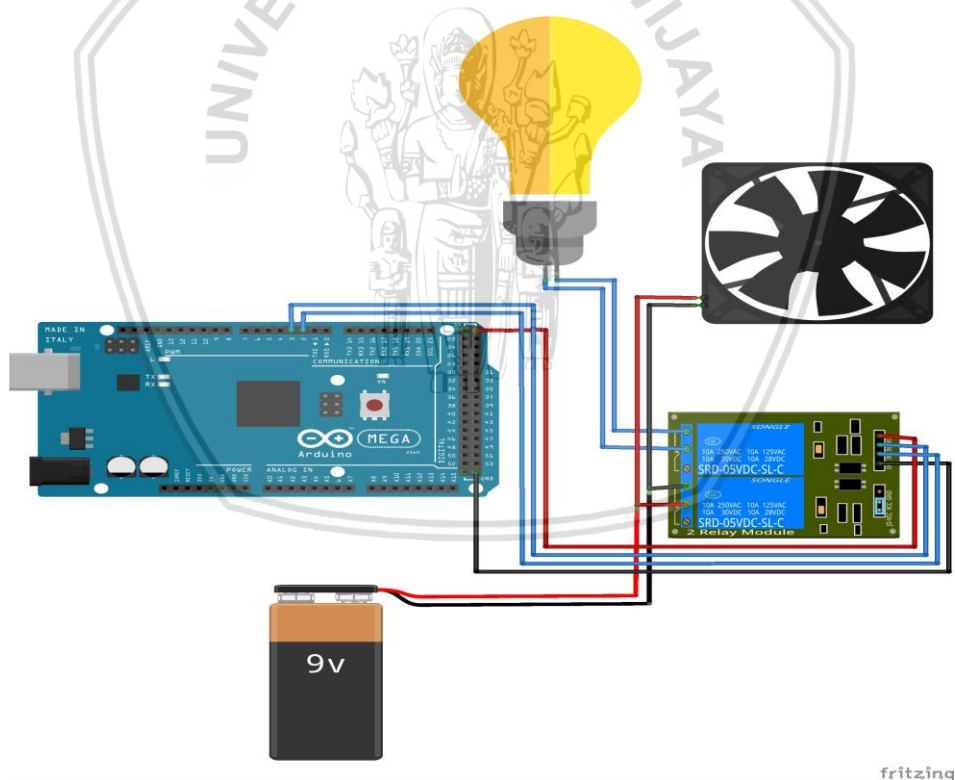
sebagai awal inisialisasi komunikasinya. Pada Arduino mega 2650, pin 10 berfungsi sebagai pin *transmitter* (Tx), sedangkan pada pin 11 berfungsi sebagai pin *receiver* (Rx).

**Tabel 5.1** Pin HC-05 ke Arduino Mega 2650

HC-05	Arduino Mega 2650
VCC	5 V
GND	GND
TX	10
RX	11

#### 5.1.2.2 Perancangan rangkaian *Relay*

*Relay* berfungsi sebagai penghubung arus DC dan arus AC yang ada pada penelitian ini. Sehingga tegangan tinggi dapat diterima oleh mikrokontroller sekaligus berfungsi sebagai saklar, untuk menghidupkan dan mematikan lampu AC dan Kipas DC. Rangkaian *Relay* ke Arduino Mega 2650 dapat dilihat pada Gambar 5.4.



**Gambar 5.4** Rangkaian *Relay* 2 Channel ke Arduino Mega 2650

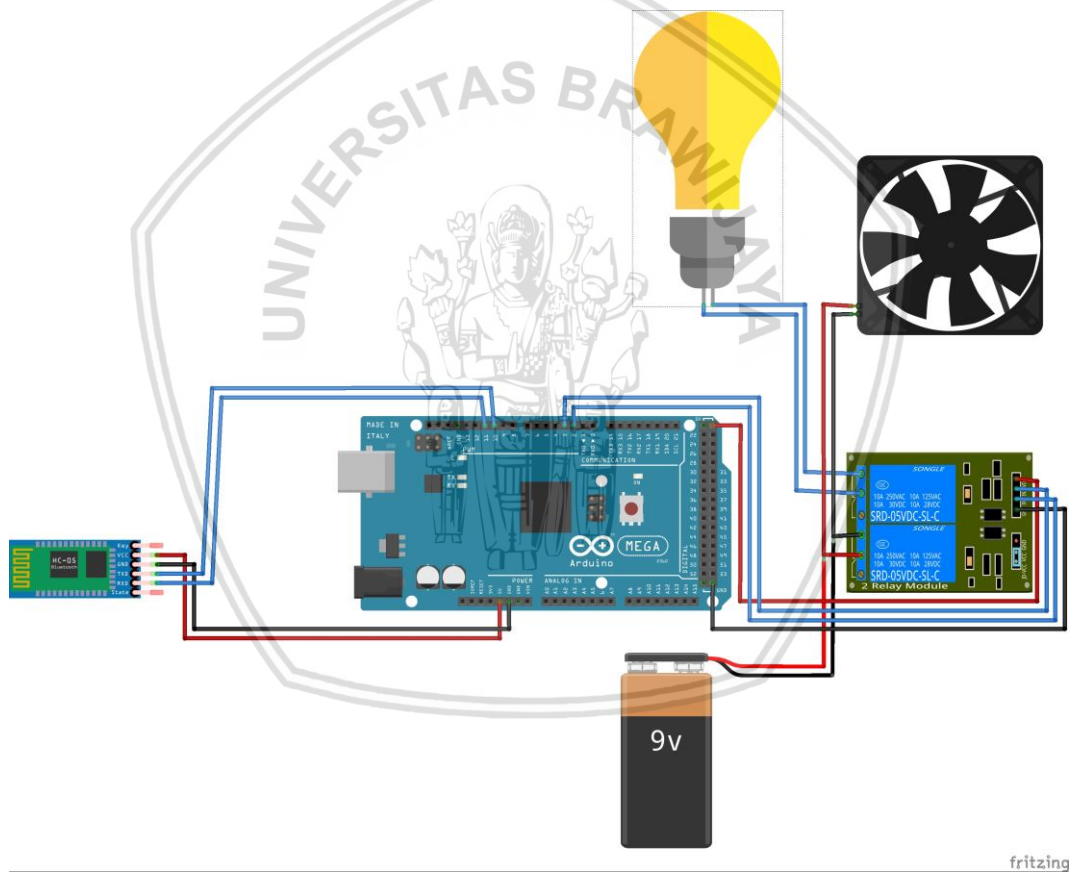
Pin *Relay* NO 1 dan NC 1 akan terhubung dengan lampu AC. Dan Pin NO2 dan NC2 untuk ke kipas DC. Untuk pin *relay* ke Arduino Mega 2650 dapat dilihat pada tabel 5.2.

**Tabel 5.2** *Relay 2 Channel* ke Arduino Mega 2650

<b>Relay</b>	<b>Arduino Mega 2650</b>
VCC	5 V
IN1	2
IN2	3
GND	GND

### 5.1.2.3 Perancangan *Hardware* secara keseluruhan

Setelah masing – masing *Hardware* dirancang dan dihubungkan menjadi kesatuan sehingga membentuk sebuah sistem yang diinginkan. Rancangan *Hardware* secara keseluruhan dapat dilihat pada Gambar 5.5.



**Gambar 5.5** Skema Rangkaian *Hardware*

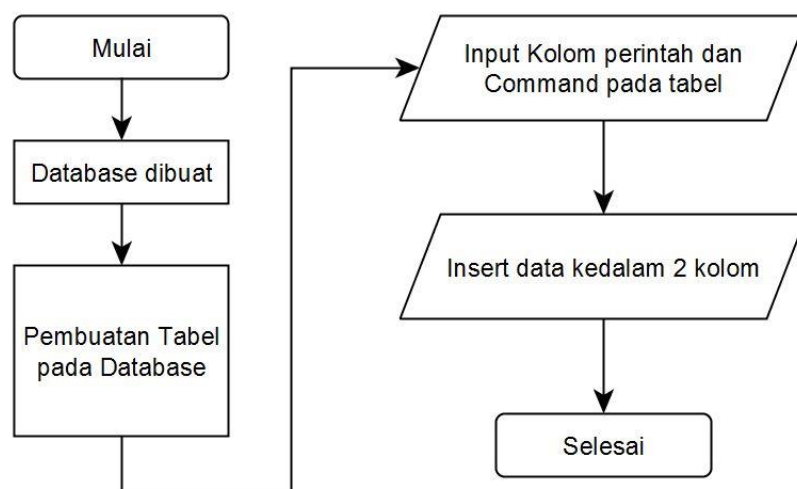
### 5.1.3 Perancangan program pada Android

Android *device* yang akan digunakan harus mempunyai koneksi *Bluetooth* untuk bisa terhubung dengan Arduino. Android *device* juga berfungsi menerima *input* suara dari *user* dan untuk melakukan proses *Text Preprocessing*, dengan menggunakan Android studio IDE sebagai sarana untuk pembuatan dan pengembangan dari *software* pengolahan suara ini. *Database* dibuat dengan menggunakan algoritme *stopword removal wordlist*.

*Database* yang ada pada Android berperan vital pada penelitian ini, karena akan mengambil kata inti dari kalimat *input* yang akan diberikan oleh *user*. Peneliti mendapatkan kata inti melalui kuesioner yang disediakan oleh *google form* secara online, dengan empat pertanyaan sesuai dengan banyaknya *output* yang disediakan oleh sistem. empat pertanyaan tersebut adalah:

1. Anda dihadapkan pada sebuah sistem (*smart home*), jika anda ingin memberikan perintah menggunakan suara secara umum (implisit) untuk menyalakan lampu maka kata seperti apa yang akan anda berikan ? contoh : hari sudah malam
2. Anda dihadapkan pada sebuah sistem (*smart home*), jika anda ingin memberikan perintah menggunakan suara secara umum (implisit) untuk mematikan lampu maka kata seperti apa yang akan anda berikan ? contoh : selamat pagi
3. Anda dihadapkan pada sebuah sistem (*smart home*), jika anda ingin memberikan perintah menggunakan suara secara umum (implisit) untuk menyalakan kipas maka kata seperti apa yang akan anda berikan ? contoh : gerah ya hari ini
4. Anda dihadapkan pada sebuah sistem (*smart home*), jika anda ingin memberikan perintah menggunakan suara secara umum (implisit) untuk mematikan kipas maka kata seperti apa yang akan anda berikan ? contoh : kok dingin ya

Setelah mendapatkan data untuk mengisi *Database*. *Database* ini dibuat menggunakan SQLite yang ada pada Android Studio. Alur pembuatan *database* akan dijelaskan pada Gambar 5.8.



**Gambar 5.6** Perancangan database

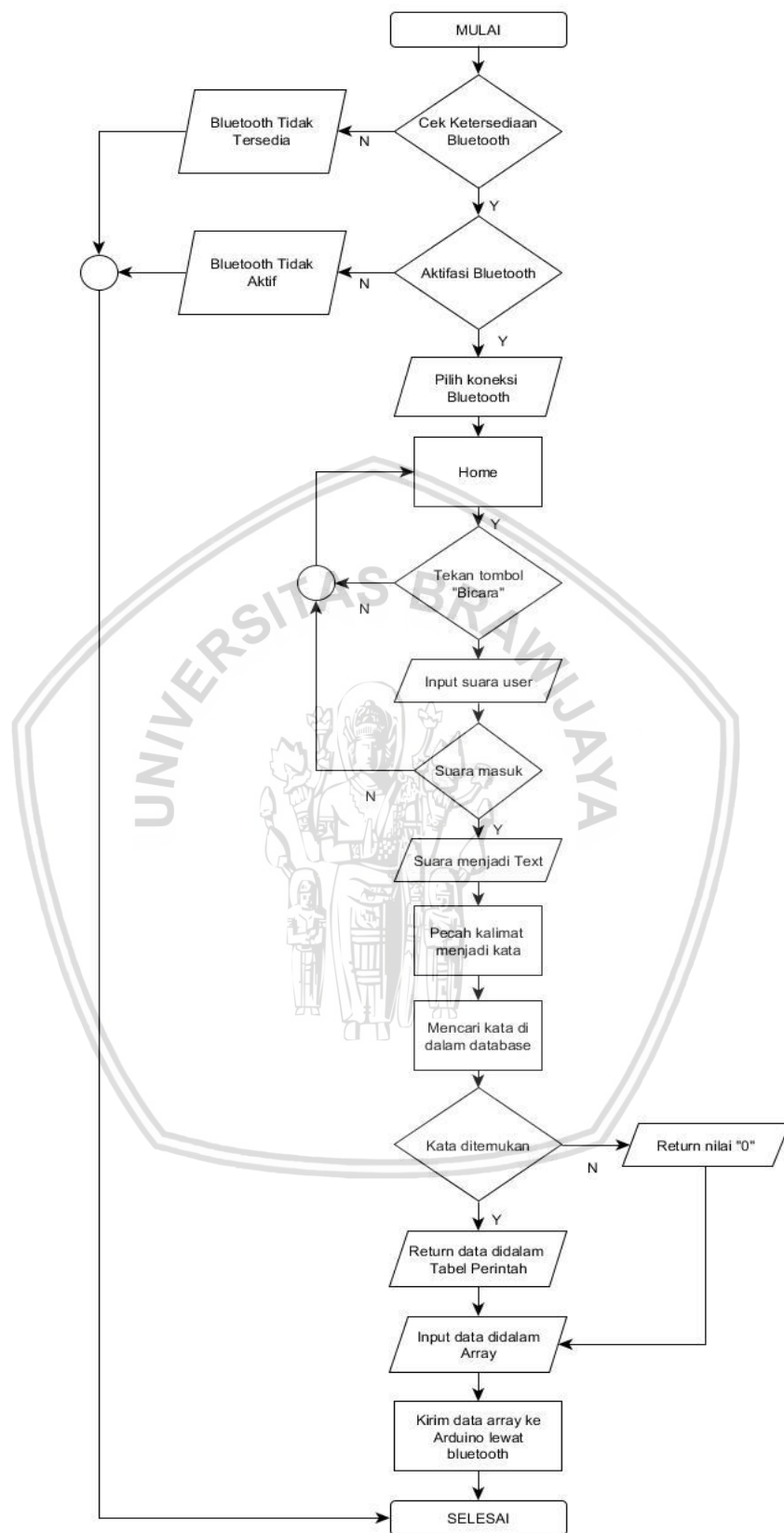
Pada Gambar 5.8 merupakan alur dari perancangan *database* yang akan digunakan. Awal mula instalasi, program dimulai dengan pembuatan *databasenya* terlebih dahulu. Setelah *database* dibentuk, kemudian dilakukan pembuatan tabel. Tabel ini memiliki dua kolom, yaitu kolom Nama dan kolom Perintah. dua kolom akan diisi dengan data kumpulan kata dan perintah. Kolom pada tabel diisi dengan perintah untuk *output*, kata hubung, kata imbuhan, dan lain-lain. Tabel 5.3 merupakan kata inti di dalam *database* yang akan digunakan.

**Tabel 5.3** Kata Inti pada *Database*

Nama	Perintah
"PANAS"	'1'
"GERAH"	'1'
"KERINGAT"	'1'
"DINGIN"	'2'
"EMBUN"	'2'
"HUJAN"	'2'
"MENDUNG"	'2'
"FLU"	'2'
"ANGIN"	'2'
"GELAP"	'3'
"MALAM"	'3'
"MENDUNG"	'3'
"BACA"	'3'
"MEMBACA"	'3'
"SILAU"	'4'
"CERAH"	'4'
"TERANG"	'4'
"PAGI"	'4'
"SIANG"	'4'

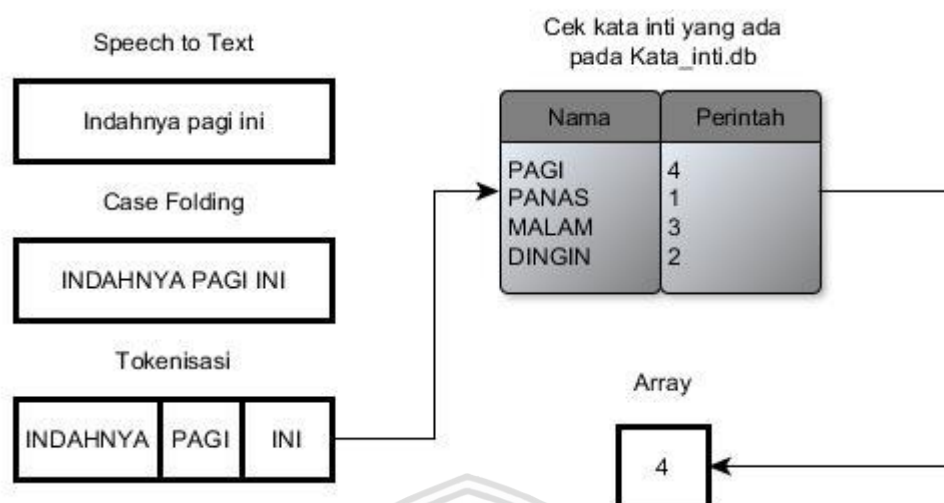
Tabel 5.3 merupakan kumpulan kata inti yang akan digunakan untuk menghasilkan *output* sistem, kolom Nama berisi kumpulan kata inti untuk *output* dari sistem, kolom ini berguna sebagai tempat mengecek apakah data pengolahan suara terdapat pada kolom ini. Kolom Perintah merupakan data hasil olahan yang nantinya dikirim ke Arduino melalui *bluetooth*. Angka – angka pada kolom Perintah mewakili *output* dari sistem pada penelitian ini.

Program ini dimulai dengan mengecek apakah Android *Device* yang digunakan ada mempunyai *Bluetooth device* atau tidak, jika tidak ada aplikasi akan tutup dengan sendirinya. Langkah selanjutnya aplikasi ini akan meminta untuk menghidupkan *Bluetooth* tersebut untuk nantinya akan menampilkan *device* yang tersedia koneksinya. Untuk terhubung dengan Arduino sistem, *user* harus memilih *Bluetooth* yang terhubung dengan Arduino, pada penelitian ini, menggunakan HC-05 sebagai *Bluetooth device*. HC-05 ini secara *default* di setting dengan mode *slave*, yaitu bisa menerima dan mengirim data. Setelah Android berhasil terhubung dengan *Bluetooth* maka akan tampil halaman *home*, pada halaman *home* ini terdapat tombol untuk memberikan perintah masukan suara. Jika tombol *speech* ini ditekan maka bisa memberikan masukan suara yang digunakan untuk perintah kontrol lampu dan kipas. Jika *user* tidak memberikan masukan suara, maka program ini akan masuk kembali ke halaman *home*. Setelah suara *user* diterima Android, akan dilakukan proses *Text Preprocessing*. Proses ini diawali dengan *Case Folding*, yaitu mengubah kalimat *input* dari *user* menjadi huruf besar semua. Selanjutnya proses *tokenisasi*, yaitu memecah kalimat tersebut menjadi kata, lalu setiap kata dicocokkan dengan kolom Nama yang ada pada database Kata\_inti.db, jika ada kata yang cocok dengan isi dari kolom Nama, maka isi dari kolom kedua yaitu kolom Perintah akan dimasukkan ke dalam *array*. Namun jika tidak ada kata yang cocok dalam *database* akan diberikan nilai '0', dan dimasukkan ke dalam *array*. Diagram alur program sistem ini dapat dilihat pada Gambar 5.9.



**Gambar 5.7** Flowchart alur program Android





**Gambar 5.8** Alur pengolahan Teks

Gambar 5.10 merupakan diagram alur pengolahan teks dari sistem. Kalimat “Indahnya pagi ini” akan diubah menjadi huruf besar terlebih dahulu, proses ini dinamakan *Case Folding*. Kemudian dipisah menjadi satuan kata “INDAHNYA”, “PAGI”, “INI”, proses ini disebut dengan tokenisasi. Kemudian kata-kata ini akan dicocokkan dengan *database*, kata “PAGI” terdapat pada kolom Nama dan menghasilkan perintah yang ada pada kolom. Perintah yaitu ‘4’. Perintah 4 ini yang dikirim ke Arduino secara sekuensial.

#### 5.1.4 Perancangan program pada Arduino

Selain perancangan *hardware*, pada Arduino membutuhkan Perancangan program / *software*. Data yang dikirim dari Android *Device* akan diolah menjadi sebuah perintah kendali lampu dan kipas. Arduino mengirim data tipe *integer* ke lampu dan kipas untuk diklasifikasikan menjadi data *array*, data *array* tersebut akan diinisialisasikan menjadi *output*. Pengelompokan data dapat dilihat pada Tabel 5.4.

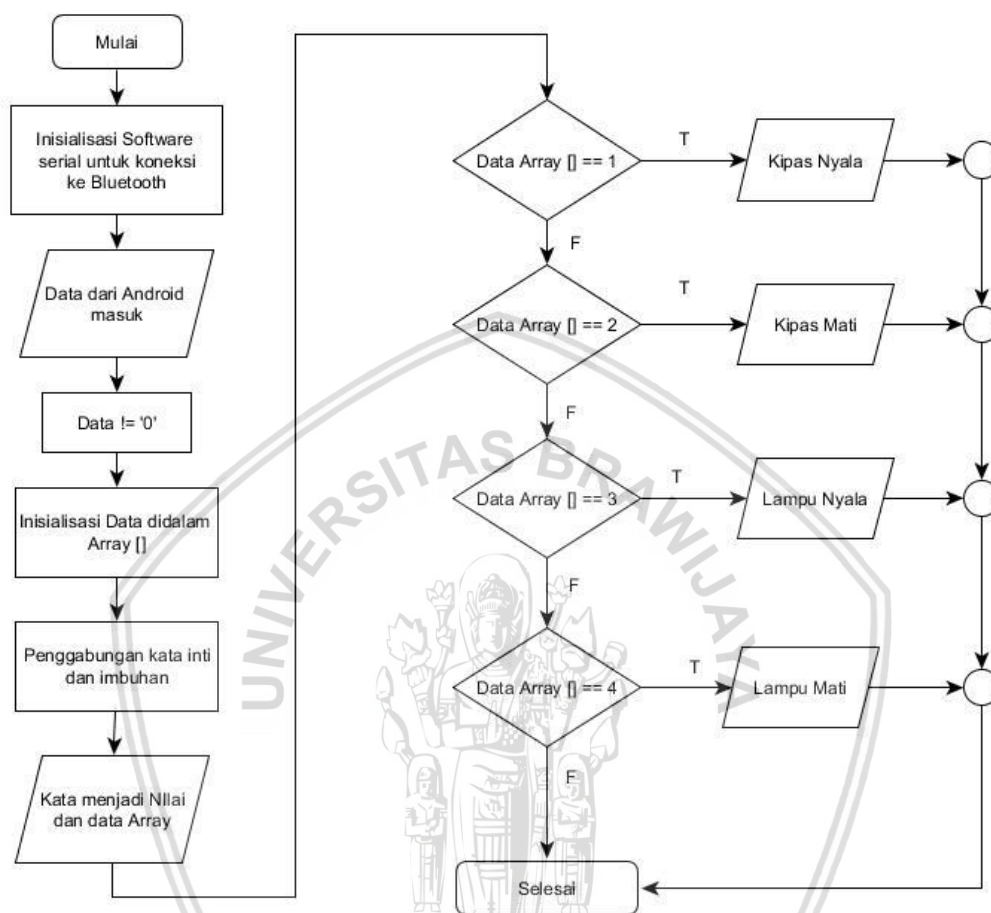
**Tabel 5.4** Perintah *Output*

Data	Perintah
1	Kipas Nyala
2	Kipas Mati
3	Lampu Nyala
4	Lampu Mati

Nilai pada Data Tabel 5.4 merupakan hasil pengolahan *text preprocessing* yang dikirim oleh Android *device* melalui koneksi *Bluetooth*. Jika data yang masuk bernilai 1, maka Arduino akan mengirimkan perintah untuk menghidupkan lampu. Jika data yang masuk bernilai 2, maka Arduino akan mengirimkan perintah untuk mematikan kipas. Jika data yang masuk bernilai 3, maka Arduino akan



mengirimkan perintah menghidupkan lampu. Jika data yang masuk bernilai 4, maka Arduino akan mengirimkan perintah mematikan lampu. Jika tidak ada kata yang cocok maka akan dikirimkan nilai 0. Alur perancangan program Arduino dapat dilihat pada Gambar 5.11.



**Gambar 5.9** Flowchart alur program pada Arduino

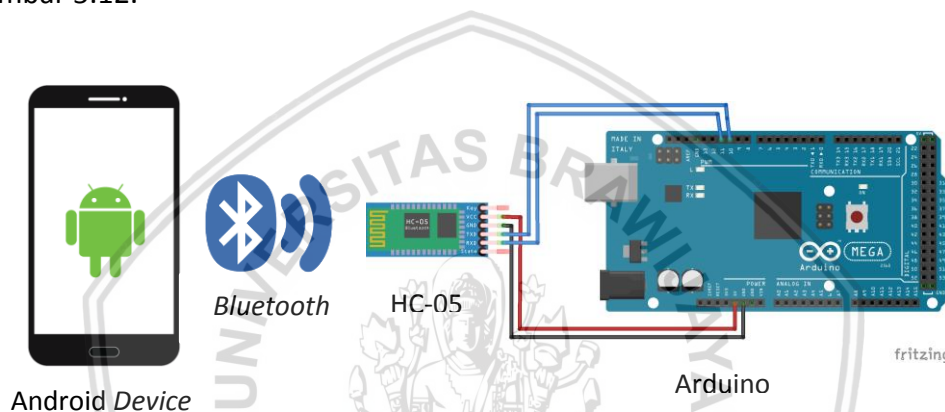
Pada Gambar 5.11 program diawali dengan memanggil *library software serial*, *library* ini digunakan sebagai inisialisasi bahwa Arduino telah terhubung dengan *module Bluetooth HC-05* melalui komunikasi serial. Data hasil *Text preprocessing* yang dikirim dari Android masuk berupa karakter, selama data yang dikirim dari Android tidak sama dengan karakter '0' maka data tersebut akan dimasukkan ke dalam *array*. Dari sekumpulan data didalam *array* akan dilakukan proses *Text Processing*, yaitu penggabungan data di dalam *array*, data yang digabung berupa kata imbuhan dan kata inti. Jika ada kata didalam *array* berbentuk imbuhan maka kata tersebut akan digabungkan dengan indeks *array* sebelum atau setelahnya. Setelah melakukan penggabungan maka akan dihasilkan nilai berupa karakter angka dengan nilai 1 sampai 4 yang nantinya diubah menjadi nilai *integer*, nilai *integer* ini merupakan perintah untuk *output* dari sistem. Nilai *integer* 1 merupakan perintah untuk kipas nyala, nilai *integer* 2 untuk kipas mati, nilai *integer* 3 untuk lampu nyala dan nilai *integer* 4 untuk lampu mati.

## 5.2 Implementasi Sistem

Bagian ini akan menjelaskan implementasi dari perancangan sistem yang telah dibuat sebelumnya, sistem ini dimulai dengan implementasi komunikasi antara tiap – tiap komponen yang saling terhubung satu sama lain sehingga menghasilkan sistem yang diinginkan. Pada bagian kedua ini akan menjelaskan bagaimana implementasi sistem pada Android. Pada bagian ketiga akan menjelaskan implementasi pada Arduino.

### 5.2.1 Implementasi komunikasi sistem

Bagian ini menjelaskan bagaimana implementasi komunikasi sistem yang akan digunakan pada penelitian ini. Implementasi komunikasi dapat dilihat pada Gambar 5.12.



**Gambar 5.10** Implementasi Komunikasi Sistem

Gambar 5.12 adalah gambaran dari implementasi perancangan komunikasi dari sistem. Android *device* tidak membutuhkan kabel untuk terhubung dengan Arduino karena jalur komunikasi yang digunakan adalah *Bluetooth*. Arduino dilengkapi dengan modul *Bluetooth* HC-05 yang disetting dengan mode *slave*, untuk dapat berkomunikasi dengan Android *device* melalui jalur *Bluetooth*.

### 5.2.2 Implementasi sistem pada Android

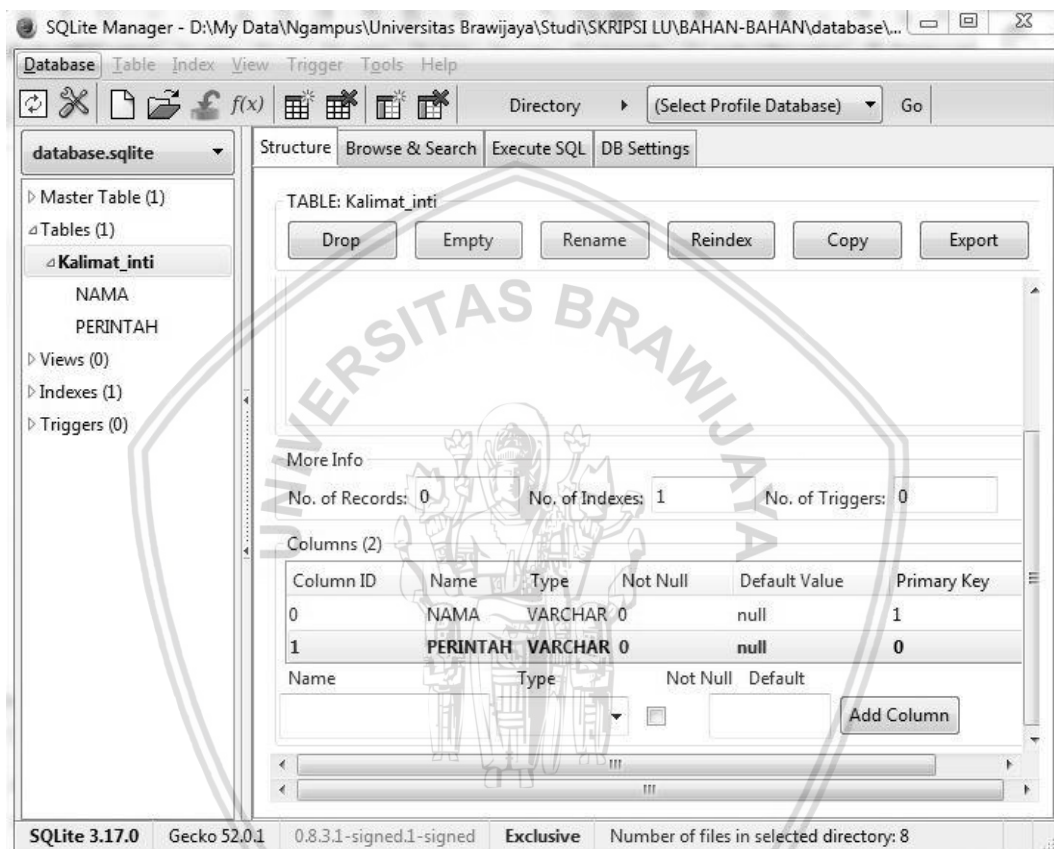
Implementasi awal program pada Android dimulai dengan pembuatan *database* dengan memanfaatkan SQLite yang ada pada Android studio, *source code* pembuatan *database* dapat dilihat pada *source code* sebagai berikut.

```

1 public static final String DATABASE_NAME = "Database.db";
2 public static final String TABLE_NAME = "Kalimat_inti";
3 public static final String COL_1 = "NAMA";
4 public static final String COL_2 = "PERINTAH";
5
6 public DatabaseHelper(Context context) {
7     super(context, DATABASE_NAME, null, 1);
8 }
9
10 public void onCreate(SQLiteDatabase db) {
11     db.execSQL("create table " + TABLE_NAME + " (NAME VARCHAR(20)
12     PRIMARY KEY, COMMAND VARCHAR(20))");
13 }

```

Pada baris pertama hingga baris ke empat merupakan inisialisasi pembuatan variable dengan tipe data *String* untuk inisialisasi pembuatan *database*. Baris ke enam berfungsi untuk memanggil fungsi dengan nama *DatabaseHelper*, fungsi ini berguna sebagai *constructor* untuk membuat *database* SQLite. Baris ke sebelas berguna untuk memanggil fungsi untuk membuat *database* saat aplikasi di *install*. Hasil dari *source code* ini dapat dilihat pada Gambar 5.13.



**Gambar 5.11** Pembuatan *database* database.db

Hasil dari pembuatan *database.db* dapat dilihat pada Gambar 5.11, terdapat 1 tabel yang bernama *Kalimat\_inti* yang berisi 2 kolom, yaitu kolom "NAMA" dan kolom "PERINTAH" dengan tipe data VARCHAR. Perintah yang digunakan untuk mengisi kolom *database* ini dapat dilihat pada *source code* ini.

```

1 public boolean insertData(String name, String command) {
2     SQLiteDatabase db = this.getWritableDatabase();
3     ContentValues contentValues = new ContentValues();
4     contentValues.put(COL_1, nama);
5     contentValues.put(COL_2, perintah);
6
7 }

```

Fungsi *source code* tersebut digunakan untuk memasukkan data ke dalam kolom NAMA dan kolom PERINTAH, baris ke empat dan 5 data dimasukkan ke dalam kolom pertama dan kedua dengan memakai fungsi *put()*. Implementasi isi dari kata inti pada *database* dapat dilihat pada Gambar 5.14.

TABLE: Kalimat_inti		Search	Show All	Add	Duplicate	Edit	Delete
rowid	NAMA	PERINTAH					
1	PANAS	1					
2	GERAH	1					
3	KERINGAT	1					
4	DINGIN	2					
5	EMBUN	2					
6	HUJAN	2					
7	MENDUNG	2					
8	FLU	2					
9	ANGIN	2					
10	GELAP	3					
11	MALAM	3					
12	BACA	3					
13	MEMBACA	3					
14	SILAU	4					
15	CERAH	4					
16	TERANG	4					
17	PAGI	4					
18	SIANG	4					

<< < 1 to 18 of 18 > >>

**Gambar 5.12** Isi dari tabel Kalimat\_inti kolom NAMA dan PERINTAH

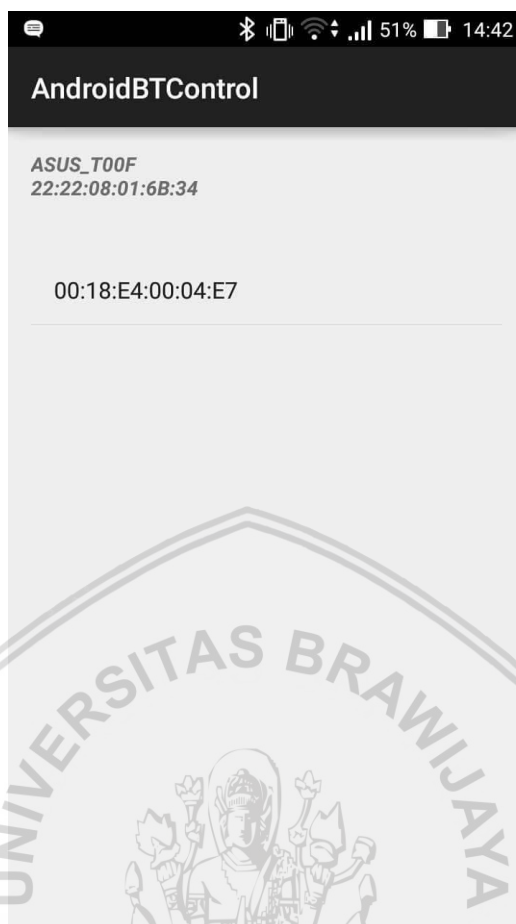
Tampilan dari aplikasi Android yang telah dibuat terdapat halaman awal dan halaman utama. Halaman awal digunakan untuk memilih koneksi *Bluetooth* supaya dapat terhubung dengan Arduino. Kode XML pada Android studio halaman awal dapat dilihat pada *source code* sebagai berikut.

```

1  <TextView
2      Android:id="@+id/info"
3      Android:textStyle="bold|italic"
4      Android:layout_width="wrap_content"
5      Android:layout_height="wrap_content"/>
6
7  <TextView
8      Android:id="@+id/status"
9      Android:textSize="28sp"
10     Android:layout_width="wrap_content"
11     Android:layout_height="wrap_content"/>
12
13 <ListView
15     Android:id="@+id/pairedlist"
16     Android:layout_width="match_parent"
27     Android:layout_height="match_parent"/>

```

Fungsi *source code* diatas akan memanggil *text view* yang berfungsi untuk menampilkan alamat dan status pada *device Bluetooth* yang sedang digunakan saat ini, selain itu juga akan menampilkan *list Bluetooth* yang terhubung dengan Android kita saat ini. Hasil XML dapat dilihat pada Gambar 5.15.



**Gambar 5.13** Halaman Awal Program

Halaman awal program ini terdapat *list Bluetooth* yang terhubung dengan *device* kita. Jika salah satu *list Bluetooth* dipilih maka akan menampilkan nama, alamat, status, *class* dari *bluetooth* yang dipilih tadi. Kode program menampilkan *list bluetooth* dapat dilihat pada *source code* berikut.

```

1 private void setup() {
2     Set<BluetoothDevice> pairedDevices =
3     bluetoothAdapter.getBondedDevices();
4     if (pairedDevices.size() > 0) {
5         pairedDeviceArrayList = new ArrayList<BluetoothDevice>();
6
7         for (BluetoothDevice device : pairedDevices) {
8             pairedDeviceArrayList.add(device);
9         }
10
11         pairedDeviceAdapter = new
12         ArrayAdapter<BluetoothDevice>(this,
13             Android.R.layout.simple_list_item_1,
14             pairedDeviceArrayList);
15         listViewPairedDevice.setAdapter(pairedDeviceAdapter);
16
17         listViewPairedDevice.setOnItemClickListener(new
18         AdapterView.OnItemClickListener() {
19
20             @Override
21             public void onItemClick(AdapterView<?> parent, View
22             view,
23                                     int position, long id) {

```



```

24         BluetoothDevice device =
25             (BluetoothDevice)
26         parent.getItemAtPosition(position);
27         Toast.makeText(MainActivity.this,
28             "Name: " + device.getName() + "\n"
29                 + "Address: " +
30         device.getAddress() + "\n"
31                 + "BondState: " +
32         device.getBondState() + "\n"
33                 + "BluetoothClass: " +
34         device.getBluetoothClass() + "\n"
35                 + "Class: " + device.getClass(),
36             Toast.LENGTH_LONG).show();
37
38         textStatus.setText("start ThreadConnectBTdevice");
39         myThreadConnectBTdevice = new
40         ThreadConnectBTdevice(device);
41         myThreadConnectBTdevice.start();
42     }
43 });
44 }
45 }

```

Pada *source code* tersebut, program diawali dengan membuat fungsi *setup()*. Pada baris ke 2 sampai ke 18, program diawali dengan menampilkan *list bluetooth* yang terkoneksi dengan Android kita. Sedangkan pada baris ke 21 sampai 41, merupakan kode untuk menampilkan nama, *address*, *state*, *class* ketika salah satu *list bluetooth* yang terhubung dipilih.

Tampilan utama pada aplikasi Android terdapat *button* yang digunakan sebagai tombol untuk mulai memasukkan *input*, selain itu terdapat tempat untuk menampilkan hasil dari *speech recognition*. Desain XML pada Android studio dapat dilihat pada *source code* sebagai berikut.

```

1  <EditText
2      Android:id="@+id/masukan"
3      Android:layout_width="match_parent"
4      Android:layout_height="wrap_content"/>
5
6  <Button
7      Android:id="@+id/send"
8      Android:layout_width="match_parent"
9      Android:layout_height="wrap_content"
10     Android:text="BICARA" />
11

```

Fungsi *source code* tersebut digunakan untuk menampilkan edit teks, dan tombol yang nantinya diberikan kode program. Hasil dari *source code* tersebut dapat dilihat pada Gambar 5.16.



**Gambar 5.14** Halaman *Home*

Pada Gambar 5.16 terdapat tombol yang berfungsi untuk memanggil *speech recognition* yang ada pada Android. Diatas tombol, terdapat tempat untuk menampilkan hasil dari *speech recognition*. Kode program untuk tombol ini dapat dilihat pada *source code* sebagai berikut.

```

1  btnSend = (Button)findViewById(R.id.send);
2
3  btnSend.setOnClickListener(new View.OnClickListener() {
4
5      @Override
6      public void onClick(View v) {
7          if(myThreadConnected!=null){
8              StartTime = SystemClock.uptimeMillis();
9              handler.postDelayed(runnable, 0);
10             btnToOpenmic();
11         }
12     });
13

```

Pada kode program tersebut, tombol “BICARA” diberikan *id* dengan nama “send”. pada baris ke 3 sampai 12 kode program ini digunakan untuk inisialisasi jika tombol di klik maka akan memanggil *speech recognition* pada Android. Contoh hasil dari pengolahan suara dapat dilihat pada gambar 5.17.





**Gambar 5.15** Implementasi dari pengolahan suara

Jika tombol “BICARA” ditekan, lalu *user* memberikan *input* suara, maka akan ditampilkan hasil dari pengolahan suara seperti 5 yang menghasilkan kalimat “contoh kata”. Angka yang berada di bawah tombol “BICARA” merupakan waktu yang dibutuhkan untuk pengolahan suara (satuak detik). Angka yang berada paling bawah merupakan waktu untuk proses *Text Preprocessing* (satuan nano detik). Hasil pengolahan suara yang diterima akan dipecah menjadi satuan kata (Tokenisasi) menggunakan perintah *split*, lalu tiap kata akan dicek di dalam *database*, kode untuk mencari kata di dalam *database* dapat dilihat pada kode sebagai berikut.

```

1  Cursor res = db.rawQuery("select COMMAND from " + TABLE_NAME + "
2  where NAME=\"" + name + "\", null);
3
4  if(res.getCount() == 0) {
5      return "0";
6  }
7
8  while (res.moveToNext()) {
9      return res.getString(0);
10 }
11 return "0";
12 }
```

Pada *source code* tersebut, semua kata hasil pemotongan kalimat (*split*) dicari didalam *database* menggunakan perintah *select*. Jika kata terdapat pada tabel Name, maka akan dikembalikan kolom Command, namun jika kata tidak terdapat pada kolom Name, maka akan diberikan nilai “0”.

### 5.2.3 Implementasi sistem pada Arduino

Pada Arduino program dimulai dengan memanggil *library software serial* untuk inialisasi bahwa *Bluetooth* HC-05 terhubung melalui komunikasi serial. Dan *Relay* yang berfungsi sebagai saklar terhadap kipas dan lampu. Implementasi Kode program dapat dilihat pada *source code* sebagai berikut.

```

1  #include <SoftwareSerial.h>
2  SoftwareSerial BT(10, 11);
3  char Command[100];
4  int totalCom = 0;
5  const int lampu = 2;
6  const int kipas = 3;
7  int mainstatus = 0;
8  int SpeechStatus = 0;
9
10 void setup() {
11     BT.begin(9600);
12     Serial.begin(9600);
13     pinMode(lampu, OUTPUT);
14     pinMode(kipas, OUTPUT);
15     digitalWrite(lampu, HIGH);
16     digitalWrite(kipas, HIGH);
17 }
18

```

Baris ke 1 sampai 6 merupakan inialisasi untuk tiap – tiap *hardware*. Arduino terhubung dengan HC-05 melalui komunikasi serial dengan *baud rate* 9600, pin 10 pada Arduino sebagai *receiver*, dan pin 11 sebagai *transmitter*. Arduino terhubung dengan *Relay* melalui pin 2 dan pin 3, pin 2 digunakan untuk kontrol lampu, pin 3 digunakan untuk kontrol kipas. Baris 11 sampai 12 merupakan inialisasi dari HC-05. Baris 13 sampai 16 merupakan inialisasi *output* dari sistem yaitu Kipas dan Lampu. Implementasi alur penerimaan data olahan dari Android dapat dilihat pada *source code* berikut.

```

1  while (BT.available()) {
2      char c = BT.read();
3      delay(10);
4      if (c != '0') {
5          Command[totalCom] = c ;
6          totalCom++;
7      }
8  }

```

Data yang masuk dari Android dimasukkan ke dalam *array*, dengan inialisasi *BT.Available()* data yang masuk dari Android dipisah dan dikumpulkan menjadi kumpulan *array*. *Array* tersebut akan diolah dengan digabungkan atau berdiri sendiri, proses ini dinamakan *array processing*. Proses pengolahan data utama pada Arduino untuk diolah menjadi perintah menghidupkan atau mematikan lampu dan kipas dapat dilihat pada *source code* berikut.

```

1  if (SpeechStatus == 1) {
2      digitalWrite(kipas, LOW);
3      Serial.println("Kipas Nyala");
4  }

```

Jika data yang diterima Arduino dari Android bernilai 1, maka Kipas akan Nyala dan akan menampilkan kalimat “Kipas Nyala” pada serial monitor Arduino.

1	<code>else if (SpeechStatus == 2) {</code>
2	<code>digitalWrite(kipas, HIGH);</code>
3	<code>Serial.println("Kipas Mati");</code>
4	<code>}</code>

Jika data yang diterima Arduino dari Android bernilai 2, maka Kipas akan Mati dan akan menampilkan kalimat "Kipas Mati" pada serial monitor Arduino.

1	<code>if (SpeechStatus == 3) {</code>
2	<code>digitalWrite(lampu, LOW);</code>
3	<code>Serial.println("Lampu Nyala");</code>
4	<code>}</code>

Jika data yang diterima Arduino dari Android bernilai 3, maka Lampu akan Nyala dan akan menampilkan kalimat "Lampu Nyala" pada serial monitor Arduino.

1	<code>else if (SpeechStatus == 4) {</code>
2	<code>digitalWrite(lampu, HIGH);</code>
3	<code>Serial.println("Lampu Mati");</code>
4	<code>}</code>

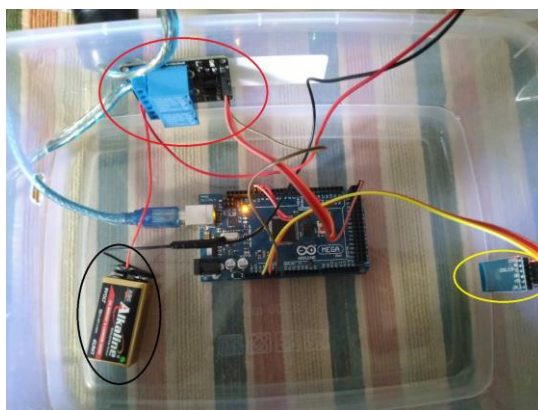
Jika data yang diterima Arduino dari Android bernilai 4, maka Lampu akan Mati dan akan menampilkan kalimat "Lampu Mati" pada serial monitor Arduino.

### 5.3 Implementasi *Hardware* Sistem

Implementasi ini merupakan tahap akhir dari implementasi *hardware* dimana *hardware* yang telah dirancang dibuat menjadi kesatuan agar dapat bekerja sesuai dengan sistem yang telah dirancang pula. *Hardware* dapat dilihat pada gambar 5.6.

Gambar 5.6 merupakan implementasi perangkat keras yang telah dirancang. *Bluetooth module* HC-05 ditandai dengan lingkaran berwarna kuning. *Relay* yang berfungsi sebagai saklar sistem ditandai dengan warna merah. Baterai untuk *supply* kipas ditandai dengan warna hitam. Sistem yang telah dirancang dimasukkan ke dalam kotak berukuran 25 x 10 cm, sedangkan untuk *output* dari sistem diletakkan diluar kotak. Dapat dilihat pada gambar 5.7.

Gambar 5.7 merupakan sistem tampak dari luar, *output* sistem yang berupa lampu dan kipas dikontrol oleh Arduino yang berada di dalam kotak. Arduino akan menerima hasil pengolahan suara dari Android melalui *Bluetooth module* HC-05.



Gambar 5.16 Implementasi *hardware*



**Gambar 5.17** Sistem tampak luar



## BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan menjelaskan mengenai pengujian yang akan dilakukan, dimulai dari prosedur dan proses pengujian yang akan menghasilkan data untuk di analisis nantinya. Proses pengujian ini dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan analisis kebutuhan yang diinginkan. Pengujian dilakukan dengan tahapan-tahapan yang telah ditentukan, dan hasil yang diperoleh dari pengujian akan dianalisis agar dapat diambil kesimpulan dari penelitian yang telah dilakukan. Pengujian pada bab ini meliputi pengujian fungsional sistem, proses pengolahan suara menjadi teks, proses pengolahan teks, efektivitas aplikasi sistem.

### 6.1 Pengujian Fungsional Sistem

#### 6.1.1 Tujuan pengujian

Pengujian ini bertujuan untuk mengetahui apakah sistem dapat berkomunikasi dengan baik dan dapat melakukan pengolahan *Text Processing*. Sistem akan diuji dari segi Android dan Arduino

#### 6.1.2 Prosedur pengujian

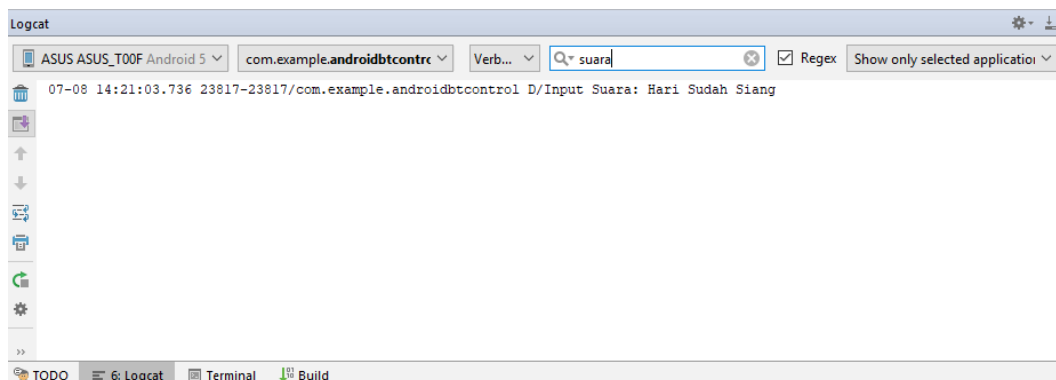
Prosedur yang dilakukan dalam pengujian ini adalah:

1. Merancang *Bluetooth Module* HC-05 dengan Arduino menggunakan kabel penghubung agar mikrokontroller dan Android dapat terhubung dengan pin yang dipakai
2. Hubungkan Android *Device* dengan komputer
3. Jalankan aplikasi dan buka Android Studio pada komputer yang berfungsi untuk melihat *console log* hasil *input* suara
4. *User* melakukan *input* suara lewat Android *Device*
5. Mengamati hasil *output* yang ada pada *console log* Android Studio dan membandingkannya dengan hasil olahan *Speech Recognition* yang ada pada Android *Device*
6. Menarik kesimpulan dari hasil pengujian

#### 6.1.3 Hasil Pengujian

Pengujian pada Android *device* dimulai dari tampilan pengolahan *speech recognition*. Jika keluaran dari Android *device* sesuai dengan yang diinginkan maka sistem akan melakukan *text splitting* dan pencocokan pada *database*. Salah satu hasil pengujian dapat dilihat pada Gambar 6.1.





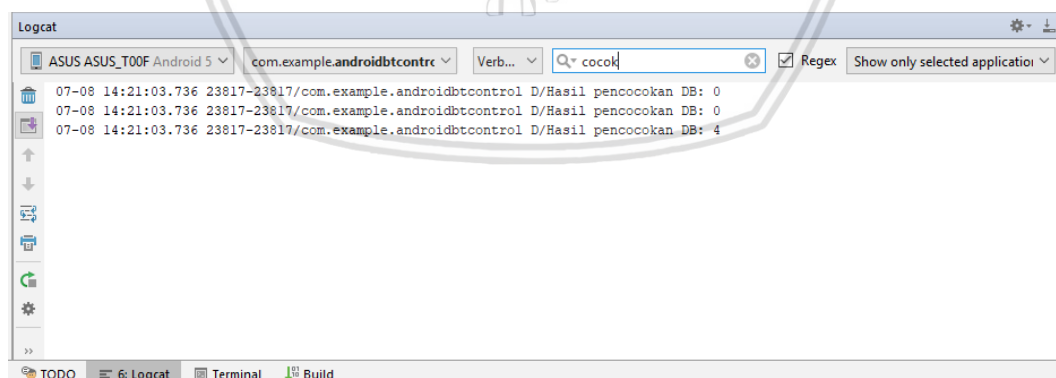
**Gambar 6.1** Hasil pengujian masukan suara

User memberikan masukan suara berupa kalimat “Hari sudah siang”, melalui *speech recognition* pada Android lalu dilakukan *splitting*, hasil pengujian pemisahan kalimat dapat dilihat pada Gambar 6.2.



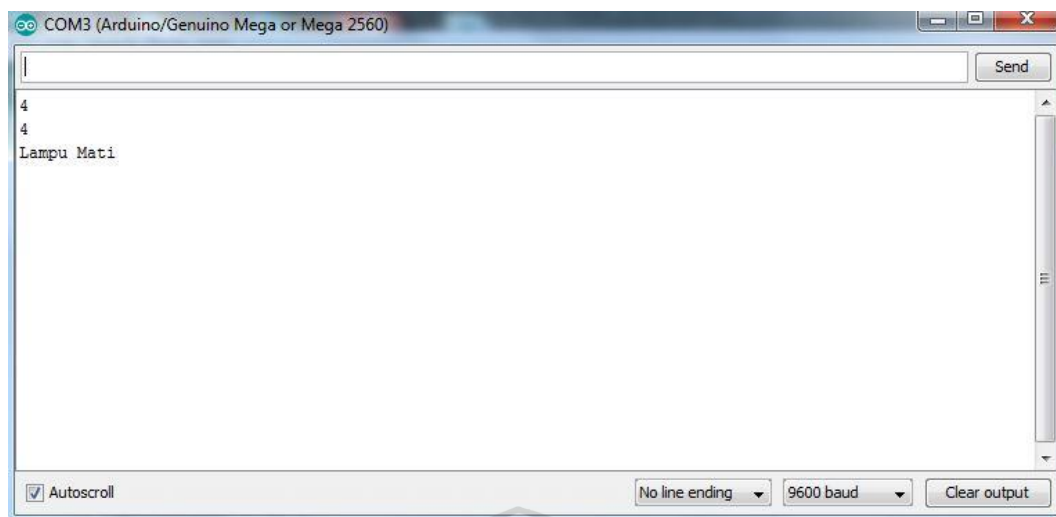
**Gambar 6.2** Hasil pengujian pemotongan kalimat menjadi kata

Pada Gambar 6.2 Kalimat masukan dari *user* dipisah menjadi satuan kata, “Hari” ,”Sudah”, “Siang”.



**Gambar 6.3** Hasil pencocokan tiap kata pada *database* di Android

Tiap satuan kata yang masuk akan dicek di dalam *database* Gambar 6.3 merupakan pengujian pengecekan tiap kata di dalam *database*, lalu dikirim ke Arduino.



**Gambar 6.4** Pengujian fungsional pada Arduino

Gambar 6.4 merupakan pengujian pengiriman data dari Android yang dapat dilihat pada serial monitor pada Arduino. Data tersebut akan diolah menjadi *output* sistem. Angka yang tampil pada serial monitor Arduino merupakan *output* dari sistem. Perintah “1” untuk Kipas Nyala, “2” untuk Kipas Mati, “3” Lampu Nyala, “4” Lampu Mati. Hasil 10 kali pengujian secara keseluruhan dapat dilihat pada Tabel 6.1.

**Tabel 6.1** Hasil Pengujian Fungsional Sistem

Banyak Kata	Kata / kalimat	Hasil Benar kata	Hasil Salah kata
1	Hai	1	0
	Pagi	1	0
	Panas	1	0
	Siang	1	0
	Gelap	1	0
	Ini	1	0
	Terang	1	0
	Cerah	1	0
	Keringat	1	0
	Malam	1	0
2	Hari panas	2	0
	Besok malam	2	0
	Pagi aja	2	0
	Sudah siang	2	0



	Aduh panas	2	0
	Terang sekali	2	0
	Dingin banget	2	0
	Silau sekali	2	0
	Panas dingin	2	0
	Cerah Sekali	2	0
3	Hari sudah siang	3	0
	Hari ini panas	3	0
	Kenapa dingin sekali	3	0
	Silau sekali ya	3	0
	Sudah malam nih	3	0
	Siang ini Panas	3	0
	Beri saya cahaya	3	0
	Dunia sudah terang	3	0
	Hari yang panas	3	0
	Kok dingin ya	3	0
4	Udah masuk angin nih	4	0
	Lampu off udah terang nih	4	0
	Sudah terang ya diluar	4	0
	Panas nih saya gerah	4	0
	Kenapa hari ini panas	4	0
	Keringat nih karena panas	4	0
	Dingin banget hari ini	4	0
	Silau banget hari ini	4	0
	Kenapa panas banget gusti	4	0
	Gelap dan dingin nih	4	0
5	Pagi yang cerah untuk kerja	5	0
	Kenapa harus panas hari ini	5	0
	Dingin banget berasa di alaska	5	0
	Keringat habis main futsal nih	5	0
	Loh udah malam saja ya	5	0

	Kenapa kamar ini panas sekali	5	0
	Cuaca hari ini dingin banget	5	0
	Kok udah gelap aja ya	5	0
	Sudah pagi saya mau kerja	5	0
	Ruangannya panas banget bos coy	5	0

Tabel 6.1 merupakan nilai hasil pengujian fungsional sistem dengan 10 kali percobaan pada setiap masukan katanya. Hasil nilai kata benar pada pengujian ini sebanyak 150 benar tanpa ada kata yang salah.

#### 6.1.4 Analisis pengujian

Hasil pengujian 10 kali yang telah dilakukan dapat dihasilkan persentase nilai benar dengan menggunakan rumus 2.1.

Hasil perhitungan sebagai berikut.

$$\text{Nilai Persentase Akurasi} = \frac{150}{150} \times 100\% = 100\%$$

Pengujian fungsional pada sistem ini dapat disimpulkan berhasil karena memiliki persentase kebenaran sebesar 100%.

## 6.2 Pengujian Proses Pengolahan Teks

### 6.2.1 Tujuan Pengujian

Pengujian ini berfungsi untuk mengetahui apakah sistem mampu mengolah kalimat yang dijadikan masukan dari *user* untuk mendapatkan kata inti.

### 6.2.2 Prosedur Pengujian

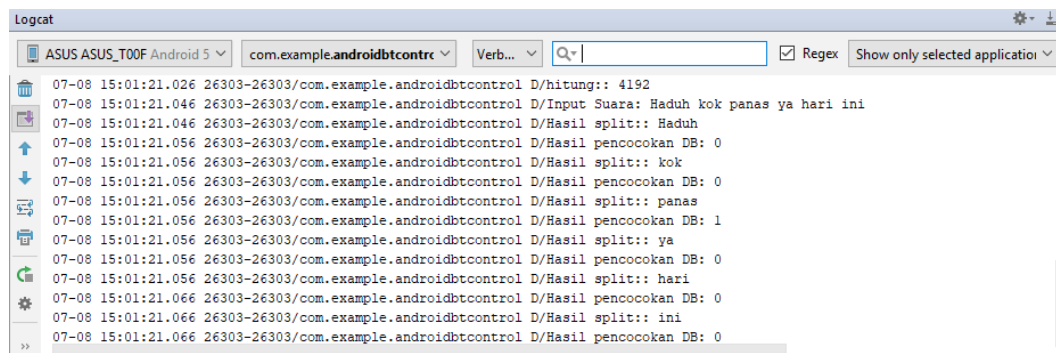
Prosedur yang dilakukan dalam pengujian ini adalah:

1. Hubungkan Android *device* dengan komputer lalu buka Android Studio
2. Buka aplikasi yang ada pada Android *device* kemudian hubungkan dengan HC-05 yang ada pada Arduino
3. *User* melakukan *input* suara ke Android *device*
4. Melihat hasil *input* suara di *console log* yang ada pada Android Studio
5. Menarik kesimpulan dari hasil pengujian

### 6.2.3 Hasil Pengujian

#### a. *User* 1

*User* 1 memberikan perintah “Haduh kok panas ya hari ini”. Hasil pengolahan kalimat dapat dilihat pada Gambar 6.5.



**Gambar 6.5** Hasil pengujian *user* pertama

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “Haduh”, “kok”, “panas”, “ya”, “hari”, “ini”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

b. *User 2*

*User 2* memberikan perintah “dingin nih hari ini benar nggak”. Hasil pengolahan kalimat dapat dilihat pada 6.6.



**Gambar 6.6** Hasil pengujian *user* kedua

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “dingin”, “nih”, “hari”, “ini”, “benar”, “nggak”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

c. *User 3*

*User 3* memberikan perintah “Selamat siang saya ingin lampu itu mati”. Hasil pengolahan kalimat dapat dilihat pada 6.7.

```

07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Input Suara: Selamat siang saya ingin lampu itu mati
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil split:: Selamat
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil split:: siang
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 4
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil split:: saya
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:19:38.846 26978-26978/com.example.androidbtcontrol D/Hasil split:: ingin
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil split:: lampu
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil split:: itu
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil split:: mati
07-08 15:19:38.856 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0

```

**Gambar 6.7** Hasil pengujian *user* ketiga

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “Selamat”, “siang”, “saya”, “ingin”, “lampu”, “itu”, “mati”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai table command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

#### d. User 4

*User 4* memberikan perintah “Halo nama saya Rizki dan hari ini panas sekali”. Hasil pengolahan kalimat dapat dilihat pada 6.8.

```

07-08 15:59:46.176 26978-26978/com.example.androidbtcontrol D/Input Suara: Halo nama saya Rizki dan hari ini panas sekali
07-08 15:59:46.186 26978-26978/com.example.androidbtcontrol D/Hasil split:: Halo
07-08 15:59:46.186 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:59:46.186 26978-26978/com.example.androidbtcontrol D/Hasil split:: nama
07-08 15:59:46.186 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:59:46.186 26978-26978/com.example.androidbtcontrol D/Hasil split:: saya
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil split:: Rizki
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil split:: dan
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 2
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil split:: hari
07-08 15:59:46.196 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:59:46.206 26978-26978/com.example.androidbtcontrol D/Hasil split:: ini
07-08 15:59:46.206 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:59:46.206 26978-26978/com.example.androidbtcontrol D/Hasil split:: panas
07-08 15:59:46.206 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 1
07-08 15:59:46.206 26978-26978/com.example.androidbtcontrol D/Hasil split:: sekali

```

**Gambar 6.8** Hasil pengujian *user* keempat

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “Halo”, “nama”, “saya”, “Rizki”, “dan”, “hari”, “ini”, “panas”, “sekali”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

#### e. User 5

*User 5* memberikan perintah “panas gua balik dari kampus tadi capek banget sekarang”. Hasil pengolahan kalimat dapat dilihat pada 6.9.

```

07-08 16:03:01.706 26978-26978/com.example.androidbtcontrol D/Input Suara: panas gua balik dari kampus tadi capek banget sekarang
07-08 16:03:01.706 26978-26978/com.example.androidbtcontrol D/Hasil split:: panas
07-08 16:03:01.706 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 1
07-08 16:03:01.706 26978-26978/com.example.androidbtcontrol D/Hasil split:: gua
07-08 16:03:01.716 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.716 26978-26978/com.example.androidbtcontrol D/Hasil split:: balik
07-08 16:03:01.716 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.716 26978-26978/com.example.androidbtcontrol D/Hasil split:: dari
07-08 16:03:01.716 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.716 26978-26978/com.example.androidbtcontrol D/Hasil split:: kampus
07-08 16:03:01.726 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.726 26978-26978/com.example.androidbtcontrol D/Hasil split:: tadi
07-08 16:03:01.726 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.726 26978-26978/com.example.androidbtcontrol D/Hasil split:: capek
07-08 16:03:01.726 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.726 26978-26978/com.example.androidbtcontrol D/Hasil split:: banget
07-08 16:03:01.736 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 16:03:01.736 26978-26978/com.example.androidbtcontrol D/Hasil split:: sekarang
07-08 16:03:01.736 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0

```

**Gambar 6.9** Hasil pengujian *user* kelima

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “panas”, “gua”, “balik”, “dari”, “kampus”, “tadi”, “capek”, “banget”, “sekarang”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

f. *User 6*

*User 6* memberikan perintah “Kenapa hari ini dingin ya”. Hasil pengolahan kalimat dapat dilihat pada 6.10.

```

07-08 15:38:32.646 26978-26978/com.example.androidbtcontrol D/Input Suara: Kenapa hari ini dingin ya
07-08 15:38:32.646 26978-26978/com.example.androidbtcontrol D/Hasil split:: Kenapa
07-08 15:38:32.646 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:38:32.656 26978-26978/com.example.androidbtcontrol D/Hasil split:: hari
07-08 15:38:32.656 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:38:32.656 26978-26978/com.example.androidbtcontrol D/Hasil split:: ini
07-08 15:38:32.656 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:38:32.656 26978-26978/com.example.androidbtcontrol D/Hasil split:: dingin
07-08 15:38:32.666 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 2
07-08 15:38:32.666 26978-26978/com.example.androidbtcontrol D/Hasil split:: ya
07-08 15:38:32.666 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0

```

**Gambar 6.10** Hasil pengujian *user* keenam

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “Kenapa”, “hari”, “ini”, “dingin”, “ya”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

g. *User 7*

*User 7* memberikan perintah “kok habis pikir kenapa hari ini dingin mulu ya”. Hasil pengolahan kalimat dapat dilihat pada 6.11.



```

07-08 15:41:40.616 26978-26978/com.example.androidbtcontrol D/Input Suara: kok habis pikir kenapa hari ini dingin mulu ya
07-08 15:41:40.616 26978-26978/com.example.androidbtcontrol D/Hasil split:: kok
07-08 15:41:40.616 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:41:40.616 26978-26978/com.example.androidbtcontrol D/Hasil split:: habis
07-08 15:41:40.616 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:41:40.616 26978-26978/com.example.androidbtcontrol D/Hasil split:: pikir
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil split:: kenapa
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil split:: hari
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil split:: ini
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil split:: dingin
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 2
07-08 15:41:40.626 26978-26978/com.example.androidbtcontrol D/Hasil split:: mulu

```

**Gambar 6.11** Hasil pengujian *user* ketujuh

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “kok”, “habis”, “pikir”, “kenapa”, “hari”, “ini”, “dingin”, “mulu”, “ya”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

#### h. *User 8*

*User 8* memberikan perintah “ruangan nggak bisa lebih dingin lagi apa”. Hasil pengolahan kalimat dapat dilihat pada 6.12.

```

07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Input Suara: ruangan nggak bisa lebih dingin lagi apa
07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Hasil split:: ruangan
07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Hasil split:: nggak
07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Hasil split:: bisa
07-08 15:44:47.546 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil split:: lebih
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil split:: dingin
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 2
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil split:: lagi
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil split:: apa
07-08 15:44:47.556 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0

```

**Gambar 6.12** Hasil pengujian *user* kedelapan

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “ruangan”, “nggak”, “bisa”, “lebih”, “dingin”, “lagi”, “apa”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

#### i. *User 9*

*User 9* memberikan perintah “hari ini gelap sekali Tuhan”. Hasil pengolahan kalimat dapat dilihat pada 6.13.



```

07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Input Suara: hari ini gelap sekali Tuhan
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil split:: hari
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil split:: ini
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil split:: gelap
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 3
07-08 15:49:14.446 26978-26978/com.example.androidbtcontrol D/Hasil split:: sekali
07-08 15:49:14.456 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:49:14.456 26978-26978/com.example.androidbtcontrol D/Hasil split:: Tuhan
07-08 15:49:14.456 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0

```

**Gambar 6.13** Hasil pengujian *user* kesembilan

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “hari”, “ini”, “gelap”, “sekali”, “Tuhan”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

j. *User 10*

*User 10* memberikan perintah “kamar terlalu silau”. Hasil pengolahan kalimat dapat dilihat pada 6.14.

```

07-08 15:51:53.416 26978-26978/com.example.androidbtcontrol D/hitung:: 435/
07-08 15:51:53.416 26978-26978/com.example.androidbtcontrol D/Input Suara: kamar terlalu silau
07-08 15:51:53.416 26978-26978/com.example.androidbtcontrol D/Hasil split:: kamar
07-08 15:51:53.416 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:51:53.416 26978-26978/com.example.androidbtcontrol D/Hasil split:: terlalu
07-08 15:51:53.426 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 0
07-08 15:51:53.426 26978-26978/com.example.androidbtcontrol D/Hasil split:: silau
07-08 15:51:53.436 26978-26978/com.example.androidbtcontrol D/Hasil pencocokan DB: 4

```

**Gambar 6.14** Hasil pengujian *user* kesepuluh

Sistem memisahkan kalimat yang *diinputkan user* menjadi satuan kata “kamar”, “terlalu”, “silau”. Lalu semua kata ini akan dicek dalam *database*, jika terdapat kata yang sama dengan *database* akan dikirimkan nilai tabel command yang selanjutnya diolah di Arduino menjadi sebuah *output*. Namun jika tidak terdapat pada *database* akan dikirimkan nilai “0”.

## 6.2.4 Analisis Pengujian

Dari hasil pengujian 10 *user* yang telah mencoba sistem ini, pemotongan kalimat untuk menghasilkan kumpulan kata memiliki hasil yang sesuai dengan harapan peneliti. Hasil penghitungan pada pengujian ini dapat dilihat pada Tabel 6.2

**Tabel 6.2** Hasil Pengujian Pengolahan Teks dan Suara

No	User ke	Masukan Kata	Banyak Kata	Jumlah Benar
1	User 1	“Haduh kok panas ya hari ini”	6	6
2	User 2	“dingin nih hari ini bener nggak”	6	6
3	User 3	“Selamat siang saya ingin lampu itu mati”	8	8

4	User 4	"Halo nama saya Rizki dan hari ini panas sekali"	9	9
5	User 5	"panas gua balik dari kampus tadi capek banget sekarang"	9	9
6	User 6	"Kenapa hari ini dingin ya"	5	5
7	User 7	"kok habis pikir kenapa hari ini dingin mulu ya"	9	9
8	User 8	"ruangan nggak bisa lebih dingin lagi apa"	7	7
9	User 9	"hari ini gelap sekali Tuhan"	5	5
10	User 10	"kamar terlalu silau"	3	3
11	User 11	"Dingin ini tidak sedingin adzab mu gusti"	6	6
12	User 12	"Saya kedinginan"	2	2
13	User 13	"Udah nggak panas"	3	3
14	User 14	"Panas dunia tidak sepanas api neraka"	6	6
15	User 15	"Sudah terang ya di luar"	5	5
16	User 16	"Matikan lampu"	2	2
17	User 17	"Hari sudah siang"	3	3
18	User 18	"Terlalu terang"	2	2
19	User 19	"Kipas on Duh gerah"	4	4
20	User 20	"udah masuk angin nih"	4	4
21	User 21	"Saya Kedinginan"	2	2
22	User 22	"Dingin bro"	2	2
23	User 23	"Selamat pagi"	2	2
24	User 24	"ah silau"	2	2
25	User 25	"panas banget coy"	3	3
26	User 26	"Panas banget nih"	3	3
27	User 27	"aku beku"	2	2
28	User 28	"Saya hanya hamba hina yg tidak pantas mendapat pencerahan"	9	9
29	User 29	"Aku pulang gelap nih di dalam rumah"	7	7
30	User 30	"hidupkan lampu"	2	2
Jumlah			136	136

Dari data yang diperoleh dapat dihitung persentase nilai akurasi *database* dan pengolahan teks menggunakan rumus 2.1.

dengan hasil perhitungan.

$$\text{Nilai Persentase Akurasi} = \frac{136}{136} \times 100\% = 100\%$$

Dari hasil penghitungan, pengujian sistem memiliki akurasi 100% Hasil pengolahan *database* juga memiliki akurasi yang baik, maka dapat disimpulkan tingkat kebenaran pengolahan teks dapat dipastikan akurat dan benar.

### 6.3 Pengujian Sistem Dengan Individu Yang Berbeda

#### 6.3.1 Tujuan pengujian

Tujuan dari pengujian ini untuk menguji efektivitas sistem dan akurasi sistem dalam mengolah kalimat implisit *user* menjadi suatu perintah kontrol terhadap lampu dan kipas. Selain itu memastikan bahwa sistem ini berjalan lancar dengan *input* suara dari *user* yang berbeda-beda.

#### 6.3.2 Prosedur pengujian

Prosedur yang dilakukan dalam pengujian ini adalah:

1. Kondisi *output* dari sistem disesuaikan terlebih dahulu dengan keinginan dari *User* sebelum melakukan *input*
2. *User* membuka aplikasi dan menghubungkan Android *device* dengan HC-05
3. *User* melakukan *input* suara lewat Android *Device*
4. Melakukan pencocokkan dengan membanding *output* yang dihasilkan oleh sistem dengan *output* yang diinginkan oleh *User*
5. Menarik kesimpulan dari hasil pengujian

#### 6.3.3 Hasil pengujian

Hasil pengujian pada sistem ini berupa 30 *user* yang berbeda, lalu tingkat benar dan kesalahan dihitung, hasil pengujian dapat dilihat pada Tabel 6.3

**Tabel 6.3** Hasil pengujian 30 *user* yang berbeda

User ke	Kalimat <i>input</i> user	Hasil yang diinginkan user	<i>Output</i> yang dihasilkan Sistem	Sesuai dengan keinginan user
1	"Kok kamarnya panas banget"	Kipas Nyala	Kipas Nyala	Ya
2	"Kamarnya dingin banget nih"	Kipas Mati	Kipas Mati	Ya
3	"Kamarnya butuh cahaya"	Lampu Nyala	Tidak ada	Tidak
4	"Ruangannya terlalu gelap"	Lampu Nyala	Lampu Nyala	Ya
5	"Siang gini kok gelap ya?"	Lampu Nyala	Lampu Nyala	Ya
6	"Silau nih lagi insomnia"	Lampu Mati	Lampu Mati	Ya
7	"Aduh gerah pulang kerja"	Kipas Nyala	Kipas Nyala	Ya
8	"Yah Hujan"	Kipas Mati	Kipas Mati	Ya
9	"Udah pagi nih"	Lampu Mati	Lampu Mati	Ya
10	"Keringetan nih gerah"	Kipas Nyala	Kipas Nyala	Ya

11	"Hore udah tidak gerah"	Kipas Mati	Kipas Mati	Ya
12	"Sudah malam ikan bobo"	Lampu Nyala	Lampu Nyala	Ya
13	"Masih siang nih tapi kaya malam"	Lampu Nyala	Lampu Nyala	Ya
14	"Keringat nih habis futsal"	Kipas Nyala	Kipas Nyala	Ya
15	"Ruangannya terlalu panas dan terlalu terang"	Kipas Nyala Lampu Mati	Kipas Nyala Lampu Mati	Ya
16	"Suhu ruangannya enak tapi terlalu gelap"	Lampu Nyala	Lampu Nyala	Ya
17	"Ruangannya terang sekali jadi tidak bisa tidur"	Lampu Mati	Lampu Mati	Ya
18	"Sepertinya ruangan terlalu dingin"	Kipas Mati	Kipas Mati	Ya
19	"Panas banget nih bos"	Kipas Nyala	Kipas Nyala	Ya
20	"Dingin dan Gelap nih"	Lampu Nyala Kipas Mati	Lampu Nyala Kipas Mati	Ya
21	"Gerah bos"	Kipas Nyala	Kipas Nyala	Ya
22	"Panas bos ini aduh"	Kipas Nyala	Kipas Nyala	Ya
23	"Sekarang lagi gelap"	Lampu Nyala	Lampu Nyala	Ya
24	"Terang banget nih"	Lampu Mati	Lampu Mati	Ya
25	"Gerah nih"	Kipas Nyala	Kipas Nyala	Ya
26	"Dingin banget nih"	Kipas Mati	Kipas Mati	Ya
27	"Ya udah malam nih"	Lampu Nyala	Lampu nyala	Ya
28	"Sudah siang nih"	Lampu Mati	Lampu Mati	Ya
29	"Panas banget di luar"	Kipas Nyala	Kipas Nyala	Ya
30	"Dingin sekali pagi ini"	Kipas Mati	Kipas Mati	Ya

Pada Tabel 6.3 dapat dilihat hasil 30 responden pada pengujian ini. Dari 30 responden, sistem berhasil mengolah kalimat implisit menjadi suatu *output* yang benar sebanyak 29 responden. 1 kalimat dari responden tidak berhasil diolah menjadi suatu *output*, karena tidak berhasil menemukan kata yang cocok dengan yang ada di *database*.

$$\text{Nilai Persentase Akurasi} = \frac{29}{30} \times 100\% = 96,67\%$$

Sistem ini memiliki akurasi sebesar 96,67%, *output* yang tidak berhasil diolah terjadi karena kalimat *input user* tidak terdapat kata yang cocok di dalam *database* Android. Namun dapat disimpulkan sistem ini memiliki akurasi yang baik.

## 6.4 Pengujian Performa Aplikasi

### 6.4.1 Tujuan pengujian

Tujuan dari pengujian ini untuk mengetahui *delay* dari sistem yang dibuat pada Android, berapa lama waktu yang dibutuhkan sejak *input* suara dari *user* masuk sampai dengan melakukan proses *Text preprocessing*.

### 6.4.2 Prosedur pengujian

Prosedur yang dilakukan dalam pengujian ini adalah :

1. Buka aplikasi dan hubungkan Android *device* dengan HC-05 yang ada pada Arduino
2. Masukkan *input* suara ke Android *device*
3. Lihat hasil waktu yang ditampilkan di aplikasi setelah selesai melakukan *input* suara
4. Menarik kesimpulan dari hasil pengujian

### 6.4.3 Hasil Pengujian

Hasil pengujian *delay* dari sistem ini berupa *timer* dengan hitungan *milisecond* dapat dilihat pada Tabel 6.4 sampai Tabel 6.8

**Tabel 6.4** Pengujian *delay* dengan masukan 1 kata

User	Kata	Waktu Pengolahan Suara (ms)	Waktu Text preprocessing (ns)
1	Pagi	3234	3869
2	Siang	3626	3644
3	Malam	3025	3289
4	Dingin	3583	3123
5	Panas	3401	3444
6	Gerah	3134	3421
7	Keringat	4023	3611
8	Silau	3132	3321
9	Cerah	3806	3543
10	Flu	3471	3012
Jumlah		34435	34176
Rata-Rata		3443,5	3417,6

Pada Tabel 6.4 merupakan pengujian *delay* dengan memberikan *input* perintah sebanyak 1 kata, dengan melakukan sebanyak 10 percobaan. Waktu yang diperoleh dari pengolahan suara memiliki rata-rata selama 3,43 detik dan rata-rata *Text preprocessing* selama 3417,6 *nano* detik.

**Tabel 6.5** Pengujian *delay* dengan masukan 2 kata

User	Kata	Waktu Pengolahan Suara (ms)	Waktu Text preprocessing (ns)
1	Panas sekali	4000	3440
2	Dingin sekali	4979	3291
3	Udah malam	4512	3711
4	Flu berat	4021	3642
5	Angin kencang	4622	3812
6	Gelap gulita	4710	4012
7	Silau sekali	4762	4112

8	Pagi besok	4212	3442
9	Siang ini	4431	3871
10	Hujan lebat	4562	3140
Jumlah		44811	36473
Rata-Rata		4481,1	3647,3

Pada Tabel 6.5 merupakan pengujian *delay* dengan memberikan *input* perintah sebanyak 2 kata, dengan melakukan sebanyak 10 percobaan. Waktu yang diperoleh dari pengolahan suara memiliki rata-rata selama 4,48 detik dan rata-rata *Text preprocessing* selama 3647,3 *nano* detik.

**Tabel 6.6** Pengujian *delay* dengan masukan 3 kata

User	Kata	Waktu Pengolahan Suara (ms)	Waktu Text preprocessing (ns)
1	Panas banget bos	4049	3320
2	Dingin sekali bro	4247	3320
3	Hujannya lebat sekali	5123	4313
4	Duh gerah nih	4312	4021
5	Ruangan ini gelap	4071	3512
6	Terang sekali lampunya	4231	3612
7	Pagi ini cerah	5412	4908
8	Sudah malam saja	4129	4618
9	Siang besok ya	4651	3421
10	Anginnya kencang sekali	5213	3894
Jumlah		49447	39002
Rata-Rata		4944,7	3900,2

Pada Tabel 6.6 merupakan pengujian *delay* dengan memberikan *input* perintah sebanyak 3 kata, dengan melakukan sebanyak 10 percobaan. Waktu yang diperoleh dari pengolahan suara memiliki rata-rata selama 4,54 detik dan rata-rata *Text preprocessing* selama 3900,2 *nano* detik.

**Tabel 6.7** Pengujian *delay* dengan masukan 4 kata

User	Kata	Waktu Pengolahan Suara (ms)	Waktu Text preprocessing (ns)
1	Gelap ya hari ini	4444	2847
2	Panas sekali aduh gerah	6012	3778
3	Anginnya kencang saya takut	6512	3512
4	Malam ini saja kerjakan	4913	4412
5	Sudah siang saja ternyata	5612	3812
6	Pagi ini cerah sekali	5022	4812
7	Ini ruangan terang sekali	4912	4676
8	Ruangan ini kok gelap	6123	3981



9	Halo dunia selamat pagi	5412	3901
10	Gerah sekali badan gua	5112	4901
Jumlah		54083	41432
Rata-Rata		5408,3	4143,2

Pada Tabel 6.7 merupakan pengujian *delay* dengan memberikan *input* perintah sebanyak 4 kata, dengan melakukan sebanyak 10 percobaan. Waktu yang diperoleh dari pengolahan suara memiliki rata-rata selama 5,4 detik dan rata-rata *Text preprocessing* selama 4143,2 *nano* detik.

**Tabel 6.8** Pengujian *delay* dengan masukan 5 kata

User	Kata	Waktu Pengolahan Suara (ms)	Waktu <i>Text preprocessing</i> (ns)
1	Kenapa hari ini panas sekali	6712	4933
2	Kenapa hari ini dingin sekali	7212	4081
3	Tidak terasa udah malam saja	6910	3971
4	Gerah ya cuaca kota malang	6636	4222
5	Kapan ya ini hujan berhenti	6422	4003
6	Silau sekali ini ruangan bro	6555	4329
7	Dingin banget ya ini ruangan	6141	4081
8	Panas sekali seperti di neraka	5903	4534
9	Kenapa saya merasa gerah sekali	6003	3998
10	Bro lu merasa dingin gak	7022	4897
Jumlah		65516	43049
Rata-Rata		6551,6	4304,9

Pada Tabel 6.8 merupakan pengujian *delay* dengan memberikan *input* perintah sebanyak 5 kata, dengan melakukan sebanyak 10 percobaan. Waktu yang diperoleh dari pengolahan suara memiliki rata-rata selama 6,55 detik dan rata-rata *Text preprocessing* selama 4304,9 *nano* detik.

#### 6.4.4 Analisis Pengujian

Dari data pengujian yang telah dilakukan, didapat rata-rata waktu yang pengolahan masukan suara sebanyak satu kata diperlukan waktu selama 3,43 detik, 2 kata masukan selama 4,48, 3 kata masukan selama 4,54, 4 kata masukan selama 5,4 detik, dan 5 kata masukan selama 6,55. Sehingga dapat dianalisis bahwa waktu yang dibutuhkan untuk pengolahan suara *input* akan bertambah sebesar  $\pm 1$  detik sesuai banyak kata yang dimasukkan ke Android.

Waktu yang dibutuhkan untuk melakukan pengolahan *text preprocessing* dengan masukan sebanyak 1 kata membutuhkan waktu 3827,7 *nano* detik, 2 kata masukan selama 3647,3 *nano* detik, waktu yang 3 kata masukan selama 3900,2 *nano* detik, waktu yang dibutuhkan untuk pengolahan 4 kata 4143,2 *nano* detik, 5 kata masukan selama 4304,9 *nano* detik. Sehingga dapat dianalisis Sedangkan waktu yang dibutuhkan untuk pengolahan *text preprocessing* akan bertambah sebesar  $\pm 200$  *nano second* seiring dengan banyaknya kata *input* dari user.

Berdasarkan 5 jenis pengujian di atas membuktikan bahwa waktu yang dibutuhkan untuk melakukan *text preprocessing* sangat cepat dan akan bertambah seiring dengan banyaknya kata yang masuk. Sedangkan waktu yang dibutuhkan untuk pengolahan suara juga akan bertambah terus menerus seiring dengan banyaknya kata *input* yang diberikan.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis yang telah dilaksanakan pada penelitian ini, maka dapat diambil kesimpulan sebagai berikut:

1. Android *device* dapat digunakan untuk pengolahan kalimat implisit menggunakan *speech recognition Google* dan melakukan *text preprocessing* dengan *database SQLite* untuk nantinya diproses menjadi kontrol terhadap kipas dan lampu. Hasil pengujian fungsional dinyatakan berhasil dengan tingkat keberhasilan 100%.
2. Untuk mengimplementasikan sistem kontrol kipas dan lampu secara implisit menggunakan suara dengan metode *Text Processing* dibutuhkan Arduino sebagai sistem kontrol dan Android untuk pengolahan kata. Sistem berhasil mengolah kalimat *input user* di Android dan mengirim hasil olahannya ke Arduino dengan tingkat keberhasilan 100%.
3. Dari percobaan 30 *user*, sistem memiliki nilai benar sebesar 29 dan memiliki nilai salah sebesar 1. Dengan nilai yang diperoleh dari pengujian pada bab sebelumnya dapat disimpulkan bahwa pengolahan suara dan kalimat implisit yang dijadikan *input* untuk kontrol terhadap lampu dan kipas memiliki akurasi yang tinggi, yaitu sebesar 96.67%

### 7.2 Saran

Terdapat beberapa saran agar sistem ini dapat dikembangkan lebih lanjut diantaranya:

1. Sebagai pengembangan lebih lanjut diharapkan penambahan *output* yang lain, tidak terbatas hanya lampu dan kipas saja.
2. Sistem dapat dikembangkan lagi menjadi metode *Text Mining*.
3. Sistem yang dibuat dapat terhubung ke jaringan Internet.

## DAFTAR PUSTAKA

- Amrutha, S. (2015). Voice Controlled Smart Home.
- Arduino. (2017). *arduino-mega-2560*. Retrieved July 19, 2017, from <http://www.arduino.org/products/boards/arduino-mega-2560>
- Chapaneri, S. V. (2012). Spoken Digits Recognition using Weighted MFCC. *International Journal of Computer Applications*, 40(3), 7.
- Faisal Baig, S. B. (2012). Controlling Home Appliances Remotely Through Voice Command. *Cornell University Library*.
- Fauzan, M. (2013). Penerapan Metode K-Means untuk Clustering Mahasiswa Berdasarkan Nilai Akademik Dengan Weka Interface Studi Kasus Pada Jurusan Teknik Informatika UMM Magelang. *Universitas Muhammadiyah Yogyakarta Online Library*.
- Fauzi, M. A. (2017). Retrieved 10 2, 2017, from <http://malifauzi.lecture.ub.ac.id/>
- Geddes, M. (2016). *Arduino Project Handbook : 25 Practical Project To Get You Started*. San Francisco: No Starch Press.
- Hudan, R. A. (2017). Sistem Kendali Berbasis Mikrokontroler Menggunakan Protoqol MQTT Pada Smarthome. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1, 445-455.
- J., D. (n.d.). *Google says Voice earch faster and more accurate*. Retrieved 9 25, 2017, from <http://www.androidauthority.com/google-voice-search-improvements-644532>
- Kisumal, H. (2010). Aplikasi Perintah Suara Pada Windows. In Zainul, & Kodhijah (Eds.), *Aplikasi Perintah Suara Pada Windows* (p. 21). Jakarta: Universitas Islam negeri Jakarta.
- Pallas, F. A. (2017). Sistem Kendali Navigasi Quadcopter Menggunakan Suara Melalui Smartphone dan Arduino Menggunakan Text Processing.
- Reichenstein7. (2012). *Arduino Modules - L298N Dual H-Bridge Motor Controller*. Retrieved 11 4, 2017, from <http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/>
- S, S. S. (2014). Speech Recognition based Control System . *2014 Third ICT International Student Project Conference (ICT-ISPC)*.
- Setyo, W. A. (2016). Desain Interaksi Aplikasi Pengendali Smart Home Menggunakan Voice Berbasis Android. *Repositori Jurnal Mahasiswa PTIK UB*, 7.
- TerryKing. (2011). *BlueTooth-HC05-HC06-Modules-How-To*. Retrieved Mei 21, 2017, from <https://arduino-info.wikispaces.com/BlueTooth-HC05-HC06-Modules-How-To>

- V., C. S. (2012). Spoken Digits Recognition using Weighted MFCC. *International Journal of Computer Applications* .
- Vibeke, A. (2017). Teaching computers to understand human language.
- Vyskovsky, A. (2015). Any Object Tracking and Following. Prague: IEEE.
- Williams, W. a. (2010). *How to use the L298N Dual H-Bridge Motor Driver*. Retrieved 11 27, 2017, from <https://www.bananarobotics.com/shop/How-to-use-the-L298N-Dual-H-Bridge-Motor-Driver>
- Xia Feng, Y. T. (2012). Internet of Things. *International Journal Of Communication Systems*, 2, 25.

